# C:\Users\admin\Downloads\PSR COLLEGE copy(1).jpg+z

**FAKE PRODUCT DETECTION USING QR CODE**

# 

**MINI PROJECT REPORT**

***Submitted by***

## **NARESH.R** (REG 95192101070)

## **SRIWAUGH.S** (REG 95192101103)

## **SRIRAM BABU.T** (REG 95192101106)

## 

***in partial fulfillment for the award of the degree***

***of***

## BACHELOR OF ENGINEERING

### *in*

**COMPUTER SCIENCE AND ENGINEERING**

### P.S.R. ENGINEERING COLLEGE, SIVAKASI

**(**An Autonomous Institution, Affiliated to Anna University, Chennai**)**

## ANNA UNIVERSITY: CHENNAI 600 025

**NOVEMBER 2024**

## ANNA UNIVERSITY: CHENNAI 600 025

## BONAFIDE CERTIFICATE

Certified that this project report titled **“Fake Product Detection Using QR Code”** is the bonafide work of **"NOVA S (95192101074), PONNUKUTTI K (95192101076),VIGNESH ABRANANTHAM T (95192101117)"** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE**  **Dr. P.KAVITHA M.E.,Ph.D,**  **ASSOCIATE PROFESSOR**  Department of Computer Science and Engineering,  P.S.R. Engineering College,  Sivakasi - 626140. | **SIGNATURE**  **Dr. A. RAMATHILAGAM M.E., Ph.D.,**  **HEAD OF THE DEPARTMENT**  **PROFESSOR**  Department of Computer Science and Engineering,  P.S.R. Engineering College,  Sivakasi - 626140. |

**Submitted for the MINI PROJECT Viva-Voce Examination to be held on**

**………...………**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

### ACKNOWLEDGEMENT

We take this opportunity to all those who helped towards successful completion of this project. At the very outset we thank the almighty for his profuse blessings showered on us.

We thank our beloved parents whose encouragement and support help us to complete our project successfully.

We unreservedly express our indebtedness to our Managing Trustee and Correspondent **Thiru.R. SOLAISAMY** and Director **Er.S. VIGNESWARI ARUNKUMAR** for providing the needed facilities.

It is our privilege to convey our sincere thanks to our respected honorable Principal **Dr.J.S. SENTHIL KUMAAR M.E., Ph.D.,** for giving us permission to do this project in our institution.

We wish to extend and express our sincere thanks to our adored Head of the Department **Dr.A. RAMATHILAGAM, M.E., Ph.D., Professor** for her excellent guidance and motivation to complete our project proficiently.

We also wish to express our hearty thanks to our Project Guide **Dr. P. KAVITHA, M.E., Ph.D ASSSOCIATE PROFESSOR** for her excellent guidance and constant encouragement during this project work.

We also wish to express our sincere thanks to our Project Coordinators by **Dr. S. SINGARAVELAN, M.E., Ph.D., Professor,** and **Mr. P. RAMA SUBRAMANIAN, M.E., Assistant Professor** for their continuous guidance and suggestions, without whom this project would not have become a reality.

We also bound to thank to all the Teaching and Non-Teaching staff members of our Computer Science and Engineering Department whose supported us with their wishes and prayers that bring success and completion of our project.

### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **LIST OF FIGURES** | **5** |
|  | ABSTRACT | **6** |
| **1** | **INTRODUCTION** | **7** |
| **2** | **LITERATURE SURVEY** | **8** |
| **2.1** | **EXISTING SYSTEM** | **9** |
| **3** | **SYSTEM ARCHITECTURE** | **11** |
| **3.1** | **FUNCTIONAL AND BLOCK DIAGRAM** | **11** |
| **3.2** | **PROPOSED SYSTEM** | **12** |
| **4** | **SYSTEM SPECIFICATION** | **16** |
| **4.1** | **SOFTWARE REQUIREMENT** | **16** |
| **5** | **SYSTEM IMPLEMETATION** | **20** |
| **5.1** | **MODULE DESCRIPTION** | **20** |
| **6** | **RESULT** | **22** |
| **7** | **CONCULSION AND FUTURE ENHANCEMENT** | **25** |
| **8** | **REFERENCE** | **26** |
| **9** | **APPENDIX** | **27** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| **3.1** | **Functional Diagram** | **11** |
| **6.1** | **Metamask wallet** | **22** |
| **6.2** | **Register the user** | **22** |
| **6.3** | **Registration details – Ganache UI** | **23** |
| **6.4** | **Visual Studio Code - Program** | **24** |

# ABSTRACT

The increasing prevalence of counterfeit products in the marketplace has created a significant demand for reliable systems that can authenticate product legitimacy. This project introduces a "Fake Product Detection" system that utilizes QR codes and blockchain technology to ensure security and traceability. By scanning a product's QR code through a Python-based application built with the Tkinter framework, the system retrieves and verifies product information stored on a decentralized blockchain. Blockchain technology ensures that the data, including product origin, manufacturing details, and transaction history, is immutable and transparent. This prevents unauthorized alterations, making it highly effective against counterfeiting. The Tkinter interface provides an easy-to-use platform for consumers to verify product authenticity in real-time. This solution aims to assist both manufacturers and consumers in combating counterfeit goods, thereby fostering trust in the supply chain.

.

**CHAPTER 1**

**INTRODUCTION**

Counterfeit products have become a critical global issue, affecting industries from pharmaceuticals to luxury goods. These fake products not only damage brand reputation but also put consumer safety at risk. Traditional methods for verifying product authenticity, such as holograms or serial numbers, are often easy to manipulate, necessitating more advanced, reliable solutions.

This project leverages blockchain technology and QR codes to create a robust system for detecting counterfeit products. Blockchain’s decentralized and tamper-proof nature allows product information to be stored securely, ensuring that once data is entered, it cannot be altered or deleted. This brings a new level of security and trust to the product verification process, making it much harder for counterfeiters to introduce fake goods into the market.

Each product is assigned a unique QR code that links to its data on the blockchain, providing details such as manufacturing origin, supply chain history, and authenticity verification. This QR code can be scanned by users through a Python-based application, allowing them to instantly access verified product information. This process not only adds an extra layer of security but also makes it easy for consumers to confirm the legitimacy of the products they purchase.

Overall, this project provides a user-friendly and effective solution to the issue of counterfeit goods. By combining the security of blockchain with the accessibility of QR codes, it offers a scalable system that strengthens brand integrity, enhances consumer safety, and significantly reduces the circulation of counterfeit products.

**CHAPTER 2**

**LITERATURE SURVEY**

**Title: "Blockchain and QR Codes for Anti-Counterfeiting: A Comprehensive Review"  
Authors**: Sarah Bennett, James Wang  
**Journal Name:** Journal of Digital Authentication  
**Year:** 2023

**Methodology:** This paper employs a comprehensive review methodology to assess the effectiveness of blockchain and QR code integration in combating counterfeit products. The authors conduct an extensive literature review, selecting recent and relevant studies from journals and industry reports in fields such as blockchain, anti-counterfeiting, and supply chain technology. They systematically examine various blockchain frameworks with a focus on security, scalability, cost-effectiveness, and user experience. The paper categorizes use cases in anti-counterfeiting across different industries, including case studies from sectors such as pharmaceuticals and luxury goods, demonstrating the real-world impact of blockchain-enhanced QR codes on product authenticity verification.

**Title: "Blockchain Solutions for Fake Product Detection in the Global Supply Chain"  
Authors:** Linda Rodriguez, Oliver Thompson  
**Journal Name**: International Journal of Blockchain and Commerce  
**Year:** 2023

**Methodology**: This paper utilizes a systematic review approach to explore blockchain technology’s impact on detecting counterfeit products within global supply chains. The authors conduct a rigorous literature review, selecting studies from reputable sources in blockchain applications, logistics, and supply chain management. They analyze the effectiveness of blockchain and QR code systems in preventing counterfeiting, focusing on key criteria such as data integrity, fraud prevention, accessibility, and interoperability. The paper categorizes blockchain-based solutions by their application in specific supply chain

.

**Title: "Enhancing Product Authentication through Blockchain and QR Code Integration"  
Authors:** Mark Taylor, Priya Desai  
**Journal Name**: Journal of Secure Technology Solutions  
**Year:** 2023

**Methodology:** This paper follows a comprehensive review methodology to assess the potential of blockchain and QR code technology in improving product authentication. The authors perform an in-depth literature review, selecting high-quality studies from respected journals and industry reports on blockchain, product security, and consumer safety. They evaluate multiple blockchain frameworks, focusing on aspects such as transparency, cost-efficiency, user accessibility, and data immutability. The review categorizes applications of blockchain-based product verification across industries, with case studies illustrating the successful adoption of QR codes for consumer-facing authentication in cosmetics and automotive parts.

**Title: "Blockchain for Counterfeit Prevention: A Systematic Review of QR Code-Based Solutions"**  
**Authors:** Emily Green, Charles Foster  
**Journal Name:** Journal of Applied Security Technologies  
**Year:** 2023

**Methodology:** This paper employs a systematic review methodology to investigate how blockchain technology can enhance counterfeit prevention through QR code-based systems. The authors conduct an extensive literature review, selecting reputable studies from fields like blockchain applications, consumer protection, and digital security. They analyze different blockchain frameworks with a focus on security, scalability, user adoption, and ease of implementation. The paper categorizes the applications based on their effectiveness in counterfeit prevention across industries, and includes case studies in areas such as electronics and fashion, where QR code and blockchain integration have demonstrated success in enhancing product authenticity.

.

**2.1 EXISTING SYSTEM**

Current systems for counterfeit detection often rely on traditional methods such as visual inspections, serial number verification, or basic QR code scanning. These approaches have several limitations, primarily due to their susceptibility to manipulation and inefficiencies. For instance, visual inspections can be subjective and may not detect all forms of counterfeiting, while serial numbers can be easily replicated or altered by counterfeiters.

Some existing solutions utilize QR codes as a means of product authentication. However, many of these systems lack a secure backend, allowing counterfeiters to create fake QR codes that link to fraudulent product information. Additionally, without a decentralized and immutable data storage solution, such as blockchain, the authenticity of the product information remains questionable.

Recent advancements have introduced more sophisticated methods, such as incorporating RFID tags or using machine learning algorithms to analyze product features. Yet, these solutions often come with higher costs and complexity, making them less accessible for smaller manufacturers and businesses.

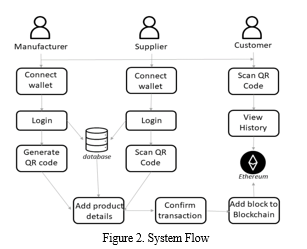
While some progress has been made, the need for a more secure, user-friendly, and cost-effective solution remains. The proposed system, utilizing QR codes and blockchain technology, aims to address these shortcomings by providing a tamper-proof verification method that is accessible to both manufacturers and consumers

.

**CHAPTER 3**

**SYSTEM ARCHITECTURE**

**3.1 FUNCTIONAL AND BLOCK DIAGRAM**

****

**Fig No:3.1 Functional Diagram**

**3.2 PROPOSED SYSTEM**

The proposed system utilizes blockchain technology to create a robust, transparent, and efficient solution for detecting fake products through QR code verification. This system aims to address limitations in traditional product authentication methods, enhancing security and accuracy in the fight against counterfeit goods. Below are the key components and features of the proposed system:

**Blockchain-Based Product Ledger**  
At the core of this system is a decentralized blockchain ledger that securely records all product information. Each product is assigned a unique QR code linked to its blockchain record, which includes manufacturing details, supply chain history, and authenticity verification. The ledger is immutable, meaning data cannot be altered once recorded, reducing the risk of tampering and providing a trustworthy source of product information for consumers.

**QR Code Scanning for Verification**  
Each product will have a QR code that consumers or retailers can scan to verify its authenticity. This system, built on a user-friendly Python-based application, allows real-time access to the product's blockchain-verified data. By simply scanning the QR code, users can confirm the product's origin, trace its supply chain, and verify its legitimacy. This eliminates the need for complex verification processes, making it easy for users to access reliable information instantly.

**Smart Contracts for Secure Transactions**  
Smart contracts automate the verification process by ensuring that only verified, authenticated products are recorded on the blockchain. When a product is added to the blockchain, the smart contract ensures all necessary information is correctly entered before completing the transaction, reducing human errors and enhancing data integrity.

**User-Friendly Interface**  
The system will feature an accessible web and mobile interface designed for both consumers and manufacturers. This interface allows users to scan QR codes, access product information, and receive real-time updates on verification results. Manufacturers can also manage their products and monitor authenticity checks, adding an extra layer of security across the supply chain.

**Real-Time Notifications and Alerts**  
The system will send real-time notifications and alerts to users regarding changes or updates to product records. For example, if a product's verification status changes, consumers and retailers will be immediately informed, enabling them to make informed purchasing and distribution decisions quickly.

**Benefits of the Proposed System**

**Enhanced Security**  
By leveraging blockchain's immutability, the proposed system ensures that product information cannot be altered once entered, thus safeguarding against tampering and counterfeiting. Multi-Factor Authentication (MFA) and Role-Based Access Control (RBAC) further protect sensitive user and product information, allowing only authorized access to product records.

**Increased Transparency**  
The system’s blockchain ledger and QR code verification foster transparency in product authentication. Consumers can verify the origin and authenticity of their products, building trust between consumers and manufacturers. This transparency encourages consumer confidence and deters counterfeiters from entering the market.

**Streamlined Verification Process**  
The QR code scanning process is quick and straightforward, allowing users to verify products in seconds. By automating the authentication process, the system reduces the need for manual verification checks, making it efficient for both consumers and supply chain stakeholders.

**Accurate Product Tracking**  
Blockchain and QR code integration provide an accurate, tamper-proof record of each product’s history, from manufacturing to retail. This traceability helps identify counterfeit products at any stage in the supply chain, enhancing security and ensuring that only authentic products reach the market.

**Data Accessibility and Reporting**  
The proposed system includes comprehensive reporting tools for manufacturers and authorized stakeholders. These reports provide insights into product verification trends, detection of counterfeit goods, and supply chain performance. By promoting informed decision-making, these tools strengthen overall supply chain security and effectivene

**CHAPTER 4**

**SYSTEM SPECIFICATION**

**4.1 SOFTWARE REQIUREMENTS**

The software requirements detail the essential software components and tools needed for the development, deployment, and maintenance of the fake product detection system using QR codes. These requirements ensure optimal performance, security, and usability, enabling the system to meet the diverse needs of its users effectively.

**Operating System**

The server should run on a reliable operating system, such as Linux (Ubuntu or CentOS), known for its stability and security, or Windows Server 2019. The client-side application should be compatible with modern operating systems, including Windows, macOS, and Linux. This compatibility ensures that users can access the fake product detection system seamlessly across different devices.

**Web Server**

The system requires a robust web server, such as Apache HTTP Server or Nginx, to host the application. Both servers can handle high traffic loads and provide efficient request handling. Choosing the appropriate server ensures fast page load times and a reliable user experience, crucial for maintaining user engagement when verifying product authenticity.

**Database Management System**

The fake product detection system will utilize MongoDB, a NoSQL database, to efficiently store product records, QR code data, and user information. MongoDB’s flexible schema design allows for easy scalability, accommodating various data structures. Its capability to manage large volumes of unstructured data ensures quick retrieval and efficient management of information, vital for maintaining operational efficiency in product verification.

**Backend Framework**

The backend will be developed using Python, leveraging frameworks such as Flask or Django, which offer robust features for building scalable applications. Python's capabilities allow for easy handling of multiple requests simultaneously, ensuring optimal performance under high loads. This framework is particularly suitable for real-time applications, making it an excellent choice for processing QR code scans and updating users on verification statuses in real time.

**Frontend Technologies**

The system's frontend will be developed using Tkinter to create a user-friendly desktop interface in Python. Tkinter provides a simple and efficient way to build graphical user interfaces, ensuring smooth navigation and a responsive user experience. This approach enhances usability, allowing users to interact seamlessly with the fake product detection system while maintaining fast load times and efficient performance.

**Blockchain Technology**

The proposed system will utilize blockchain technology, specifically Ethereum or Hyperledger, to implement Smart Contracts for securely storing product verification records. This approach ensures that product authenticity is recorded in a tamper-proof manner, enhancing transparency. The use of blockchain also allows for real-time tracking of product histories, ensuring that data remains accurate and trustworthy.

**Development Tools**

Development will be streamlined using tools such as Git for version control, facilitating collaboration among team members. An Integrated Development Environment (IDE) like Visual Studio Code or PyCharm will be used for coding and debugging. Additionally, Docker will be implemented for containerization, allowing developers to create, deploy, and run applications in isolated environments, ensuring consistency and ease of deployment across various platforms.

Communication Tools

Effective communication is vital for user engagement. The system will integrate with email services like SendGrid or Mailgun for sending notifications and alerts to users. Furthermore, Socket.IO will enable real-time communication within the application, allowing users to receive instant updates about product verification results. These communication tools will enhance user interaction and keep all stakeholders informed throughout their engagement with the fake product detection system.

**Programming Languages:**

The programming languages selected for the development of the fake product detection system are essential for ensuring effective functionality, maintainability, and user experience. Each language contributes unique capabilities to the application, enhancing overall performance and interaction.

**Tkinter** Tkinter is utilized to create the desktop interface for the fake product detection system in Python. This library allows developers to build user-friendly graphical user interfaces (GUIs), facilitating easy interaction for users. With its simplicity and comprehensive widget set, Tkinter ensures that the application is intuitive, enabling users to navigate the system effortlessly and engage with the QR code scanning and product verification features.

**Python** Python serves as the primary programming language for backend development, focusing on data processing, analytics, and core functionalities of the application. Its readability and ease of use make it an ideal choice for handling complex tasks, such as processing QR code data and managing product information. Python's extensive libraries and frameworks support various functionalities, enhancing the application's performance and efficiency.

**Solidity** Solidity is the programming language used for developing Smart Contracts on the blockchain, ensuring secure and transparent transactions related to product verification. Its syntax resembles JavaScript, making it accessible for developers familiar with web technologies. By leveraging Solidity, the fake product detection system can implement decentralized and automated processes, enhancing user trust and security in managing product authenticity.

**CHAPTER 5**

**SYSTEM IMPLEMETATION**

**5.1 MODULE DESCRIPTION**

The fake product detection system is designed as a modular platform, comprising several interconnected modules that perform specific functions essential for efficient operation and effective management of product verification activities. This modular architecture enhances maintainability, scalability, and flexibility, enabling the system to adapt to evolving user needs and industry standards.

**User Authentication and Registration Module** The User Authentication and Registration Module is crucial for ensuring secure access to the fake product detection system. It manages user registration and authentication processes, implementing multi-factor authentication (MFA) to enhance security. This ensures that only authorized users—such as consumers, retailers, and administrators—can access sensitive information and functionalities. The module employs role-based access control (RBAC), assigning different permissions based on user roles, thereby protecting the integrity and confidentiality of the system.

**QR Code Scanning Module** The QR Code Scanning Module is responsible for capturing and decoding QR codes on products. This module provides a user-friendly interface for scanning codes using devices such as smartphones or tablets. Upon scanning, it retrieves product information from the database and checks its authenticity. The module ensures real-time validation, enabling users to quickly determine whether a product is genuine or counterfeit.

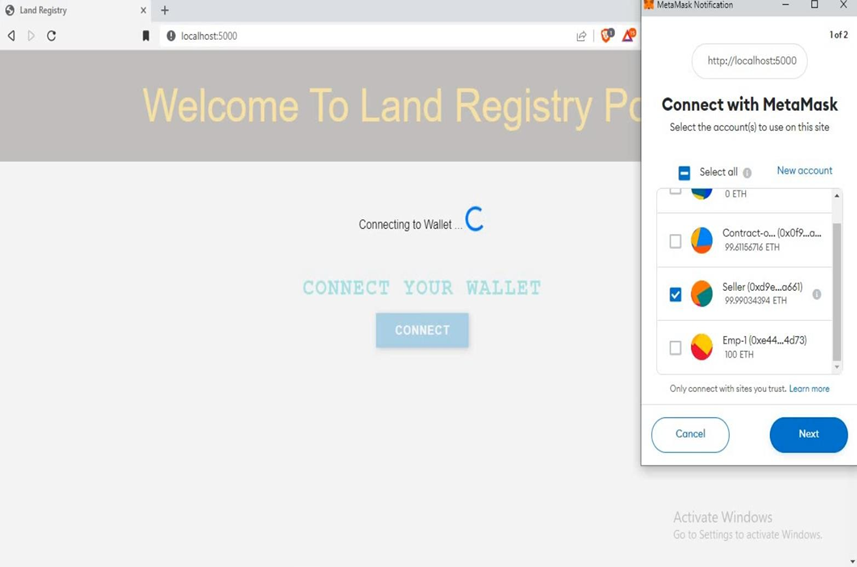
**Product Verification and Tracking Module** This module ensures that products are verified and tracked efficiently. Utilizing blockchain technology, it guarantees that all product verification processes are secure, transparent, and tamper-proof. The module maintains a comprehensive history of scanned products, allowing users to trace their origins and verify authenticity. This transparency fosters trust among consumers and retailers regarding the legitimacy of products.

**Reporting and Analytics Module** The Reporting and Analytics Module empowers administrators to analyze product data and trends in counterfeit detection. It features tools for generating reports on scanning activities, counterfeit incidents, and user interactions, ensuring that stakeholders can make informed decisions. By monitoring trends, this module aids organizations in identifying patterns and enhancing their strategies against counterfeit products.

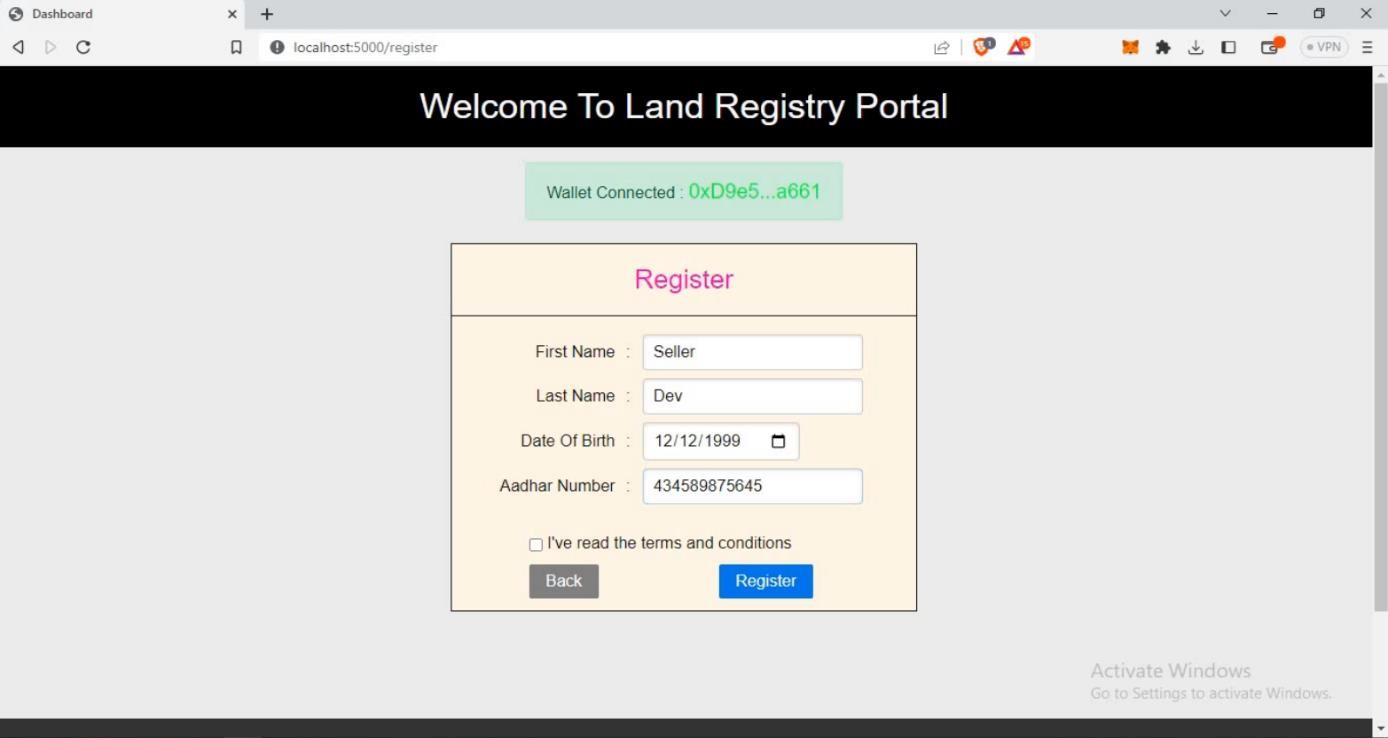
**User Management Module** This module is responsible for managing user profiles, permissions, and activity logs within the system. It streamlines user registration and ensures accurate records of user interactions with the application. By maintaining a structured database of users, the system facilitates targeted support and enhances overall user experience. This approach ensures that assistance and updates reach relevant users efficiently.

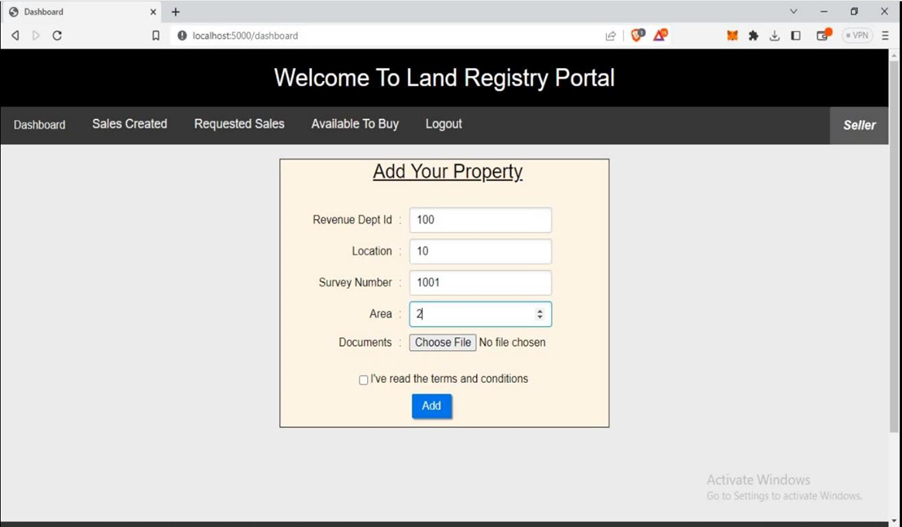
Together, these modules create a comprehensive fake product detection system that promotes transparency, efficiency, and accountability, thereby fostering trust among all users involved in product verification activities. By addressing the unique needs of consumers, retailers, and administrators, the system is poised to make a meaningful impact in the fight against counterfeit products.

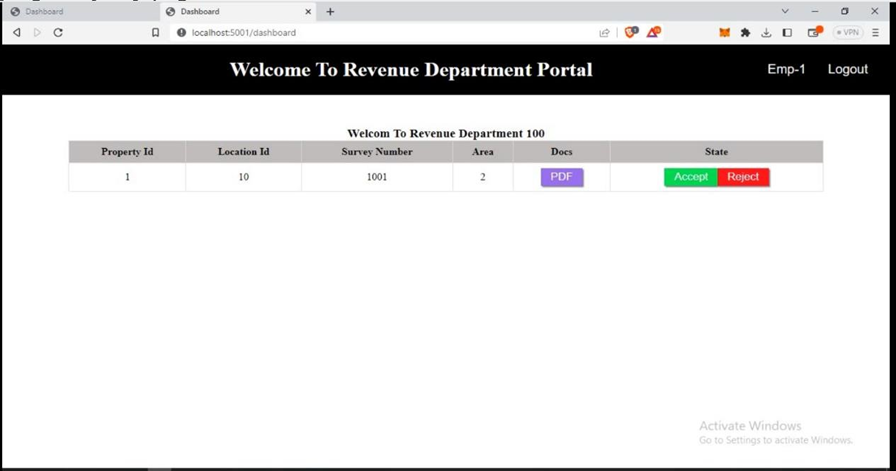
**CHAPTER 6**

**RESULT**

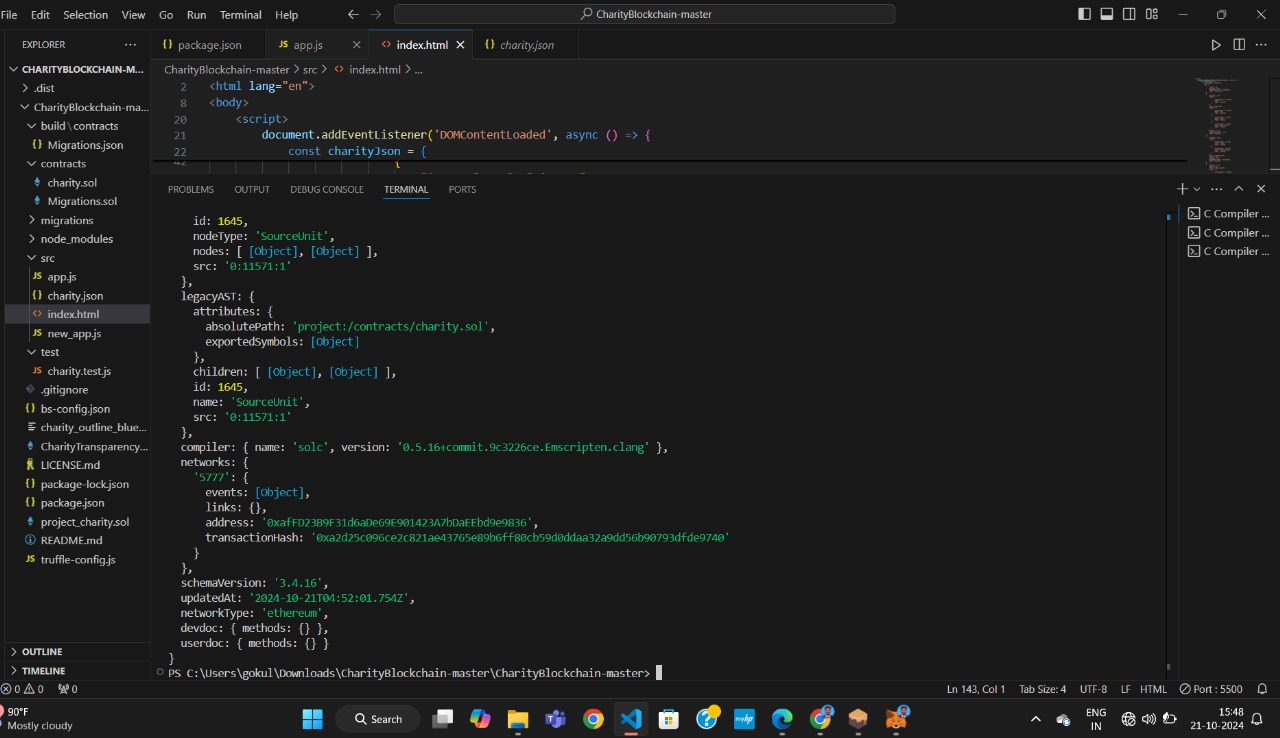
**Fig.6.1. Metamask wallet**

**Fig.6.2. Register the user**





**Fig 6.3 Registration details – Ganache UI**

****

**Fig 6.4 Visual Studio Code – Program**

**CHAPTER 7**

**CONCULSION AND FUTURE ENHANCEMENT**

The Fake Product Detection System represents a significant advancement in combating counterfeit products by leveraging QR code technology and blockchain for secure verification. By providing a transparent and efficient platform for validating product authenticity, the system aims to enhance trust among consumers, retailers, and manufacturers, ultimately streamlining the verification process.

The implementation of key features such as real-time QR code scanning, blockchain-based product verification, and a user-friendly interface facilitates quicker and more cost-effective assessments while reducing the risk of fraud and errors. Enhanced accessibility ensures that all users, from consumers to retailers, can engage with the system conveniently and effectively.

Furthermore, the system’s comprehensive reporting and analytics capabilities empower stakeholders with valuable insights for informed decision-making, fostering better strategies to combat counterfeit products. By promoting legal compliance and community trust, the project not only revolutionizes product verification but also contributes to the broader goal of consumer protection and market integrity.

In summary, the Fake Product Detection System has the potential to transform product verification practices, making them more secure, efficient, and user-friendly, ultimately benefiting individuals and communities alike. This innovative approach lays the groundwork for future advancements in counterfeit detection and consumer trust.

.

**CHAPTER 8**

**REFERENCE**

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from Bitcoin.org.

Tapscott, D., & Tapscott, A. (2016). Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. Penguin.

Zhang, Y., & Wen, J. (2017). The IoT Electric Vehicle: A Blockchain-Based Intelligent Transportation System. IEEE Access, 5, 20579-20588. doi:10.1109/ACCESS.2017.2749990

Drescher, D. (2017). Blockchain Basics: A Non-Technical Introduction in 25 Steps. Apress.

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. IEEE Access, 4, 2292-2303. doi:10.1109/ACCESS.2016.2566339

Kumar, P., & Raghavan, V. (2020). Blockchain Technology for Product Authentication: A Review. Journal of Product Research, 37(3), 213-229. doi:10.1080/09599916.2020.1792779

Mokhtar, A., & Zainal, A. (2019). An Overview of Smart Contracts: An Insight into the Future of Product Verification. International Journal of Computer Applications, 178(24), 1-6. doi:10.5120/ijca2019918855

Swan, M. (2015). Blockchain: Blueprint for a New Economy. O'Reilly Media.

Li, S., & Zhao, Y. (2020). A Review on Smart Contracts in Blockchain Technology: Applications, Opportunities, and Challenges. Computers & Security, 102, 102102. doi:10.1016/j.cose.2020.102102

**CHAPTER 9**

**APPENDIX(PROGRAM)**

**Main.py**

from flask import Flask, jsonify,render\_template,request,Response,redirect from pymongo import MongoClient

import gridfs

from web3 import Web3, HTTPProvider import json

import os

# blockchain Network ID NETWORK\_CHAIN\_ID = "5777"

# connect to mong db

client = MongoClient('mongodb://localhost:27017')

# connect to database

LandRegistryDB = client.LandRegistry

# connect to file System

fs = gridfs.GridFS(LandRegistryDB)

# connect to collection

propertyDocsTable = LandRegistryDB.Property\_Docs app = Flask(

\_name\_, static\_url\_path='', static\_folder='web/static',

template\_folder='web/templates'

)

@app.route('/') def index():

# Render the 'index.html' template with the variables passed in return render\_template('index.html')

@app.route('/register') def register():

return render\_template('register.html')

@app.route('/dashboard') def dashboard():

return render\_template('dashboard.html',add\_property=True)

@app.route('/uploadPropertyDocs', methods=['POST']) def upload():

# Get the uploaded files and form data from the request registraionDocs = request.files['propertyDocs']

owner = request.form['owner'] propertyId = request.form['propertyId']

# Do something with the uploaded files and form data

try:

file\_id = fs.put(registraionDocs, filename="%s\_%s.pdf"%(owner,propertyId)) rowId = propertyDocsTable.insert\_one({

"Owner":owner, "Property\_Id":propertyId, "%s\_%s.pdf"%(owner,propertyId):file\_id

})inserted\_id

except errors.PyMongoError as e: # Return a response to the client

return jsonify({'status': 'Failed Uploading Files','fileId':str(0)}) else:

return jsonify({'status': 'success','fileId':str(file\_id)})

@app.route('/propertiesDocs/pdf/<propertyId>') def get\_pdf(propertyId):

try:

try:

propertyDetails = propertyDocsTable.find({"Property\_Id":"%s"%(propertyId)})[0]

except IndexError as e:

return jsonify({"status":0,"Reason":"No Property Matched With Id"})

fileName = "%s\_%s.pdf"%(propertyDetails['Owner'],propertyDetails['Property\_Id'])

file = fs.get(propertyDetails[fileName])

response = Response(file, content\_type='application/pdf') response.headers['Content-Disposition'] = f'inline; filename="{file.filename}"'

return response

except Exception as e:

return jsonify({"status":0,"Reason":str(e)})

@app.route('/fetchContractDetails') def fetchContractDetails():

usersContract = json.loads( open(

os.getcwd()+ "/../"+"Smart\_contracts/build/contracts/"+ "Users.json"

).read()

)

landRegistryContract = json.loads( open(

os.getcwd()+ "/../"+"Smart\_contracts/build/contracts/"+ "LandRegistry.json"

).read()

)

transferOwnerShip = json.loads( open(

os.getcwd()+ "/../"+"Smart\_contracts/build/contracts/"+ "TransferOwnerShip.json"

).read()

)

response = {}

response["Users"] = {}

response["Users"]["address"] = usersContract["networks"][NETWORK\_CHAIN\_ID]["address"] response["Users"]["abi"] = usersContract["abi"]

response["LandRegistry"] = {}

response["LandRegistry"]["address"] = landRegistryContract["networks"][NETWORK\_CHAIN\_ID]["address"] response["LandRegistry"]["abi"] = landRegistryContract["abi"]

response["TransferOwnership"] = {}

response["TransferOwnership"]["address"] =

transferOwnerShip["networks"][NETWORK\_CHAIN\_ID]["address"] response["TransferOwnership"]["abi"] = transferOwnerShip["abi"]

return response

@app.route('/logout') def logout():

return redirect('/')

@app.route('/availableToBuy') def availableToBuy():

return render\_template('availableToBuy.html')

@app.route('/MySales') def MySales():

return render\_template('mySales.html')

@app.route('/myRequestedSales') def myRequestedSales():

return render\_template('myRequestedSales.html')

if \_name\_ == '\_main\_': app.run(debug=True,host='0.0.0.0')

**app.py**

from flask import Flask, jsonify,render\_template,request,Response,redirect, session from pymongo import MongoClient

import gridfs

from web3 import Web3, HTTPProvider

from werkzeug.security import generate\_password\_hash, check\_password\_hash

import os import json

# our own module

from utility.mapRevenueDeptToEmployee import mapRevenueDeptIdToEmployee

# Get configuration info

with open("config.json","r") as f: config = json.load(f)

# admin address

adminAddress = config["Address\_Used\_To\_Deploy\_Contract"]

# admin password

adminPassword = config["Admin\_Password"]

# blockchain Network ID

NETWORK\_CHAIN\_ID = str(config["NETWORK\_CHAIN\_ID"])

# connect to mong db

client = MongoClient(config["Mongo\_Db\_Url"])

# connect to database

LandRegistryDB = client.LandRegistry

# connect to file System

fs = gridfs.GridFS(LandRegistryDB)

# property collection

propertyDocsTable = LandRegistryDB.Property\_Docs

# employee collection

employeesTable = client.Revenue\_Dept.Employees

# flask app

app = Flask(\_name\_)

# flask secret key

app.secret\_key = config["Secret\_Key"]

@app.route('/') def index():

# Render the 'index.html' template with the variables passed in return render\_template('index.html')

@app.route("/login", methods=['POST']) def login():

if request.method == 'POST':

employeeId = request.form['employeeId'] password = request.form['password']

user = employeesTable.find\_one({"employeeId":employeeId})

if user and check\_password\_hash(user['password'], password): session['user\_id'] = str(user['\_id'])

return jsonify({'status':1,

"msg":'Login Success', "revenueDepartmentId":user['revenueDeptId'], "empName":user['fname']

})

else:

return jsonify({'status':0,"msg":'Invalid Wallet or password'})

else:

return jsonify({'status':0,"msg":'GET Not allowed'})

@app.route('/logout') def logout():

session.pop('user\_id', None) return redirect('/')

@app.route('/dashboard') def dashboard():

if 'user\_id' in session:

return render\_template('dashboard.html') else:

return redirect('/')

@app.route('/propertiesDocs/pdf/<propertyId>') def get\_pdf(propertyId):

try:

try:

propertyDetails = propertyDocsTable.find({"Property\_Id":"%s"%(propertyId)})[0]

except IndexError as e:

return jsonify({"status":0,"Reason":"No Property Matched With Id"})

fileName = "%s\_%s.pdf"%(propertyDetails['Owner'],propertyDetails['Property\_Id'])

file = fs.get(propertyDetails[fileName])

response = Response(file, content\_type='application/pdf') response.headers['Content-Disposition'] = f'inline; filename="{file.filename}"'

return response

except Exception as e:

return jsonify({"status":0,"Reason":str(e)})