Department of Information Technology
NBA Accredited

A.P. Shah Institute of Technology

G. B. Road, Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year 2022-2023

A Project Report on

Eshop: Comprehensive Blockchain based Web Framework for E-commerce

Submitted in partial fulfillment of the degree of

INFORMATION TECHNOLOGY

By
Sakshi Shinde(19104055)
Anurag Singh(19104003)
Shubhangi Lanke(19104064)

Under the Guidance of
Prof. Manjusha Kashilkar

1. Project Conception and Initiation

1.1 Abstract

- Counterfeit products play an important role in product manufacturing industries. This affects the companies name, sales, and profit of the companies.

- Blockchain technology can used for identification of real products and detect fake products.

- To ensure the identification of real products throughout the supply chain, a functional blockchain technology is used for preventing product counterfeiting.

1.2 Objectives

- To ensure everyone's access to original and safe products on portal.
- To improve the product supply chain ecosystem's trust and transparency from manufacturer to consumers.
- To secure product details using a QR code.
- To prevent consumer from receiving counterfeit products.

1.3 Literature Review

Sr. No.	Title	Key Findings	Year
1.	A Survey of Counterfeit Product Detection	Counterfeit products are growing exponentially with the enormous amount of online and black market. This paper discusses various techniques for identifying counterfeit products.	2019
2.	A Blockchain-Based Supply Chain Quality Management Framework	The proposed framework provides a theoretical basis for intelligent quality management of the supply chain based on blockchain technology. It provides a foundation to develop theories about information resource management in distributed, virtual organizations.	2019
3.	A Blockchain-Based Application System for Product Anti-Counterfeiting	A decentralized Blockchain system with products anti-counterfeiting, in that way manufacturers can use this system to provide genuine products without having to manage direct-operated stores, which can significantly reduce the cost of product quality assurance.	2020

1.4 Problem Definition

To ensure the identification of real products throughout the supply chain on E-commerce platform by using a functional blockchain technology for preventing product counterfeiting.

1.5 Scope

- Can be useful for building transparency throughout the supply chain.
- Can be managed and controlled by different admins.
- Can be used by any new user because of the adoptability of the process.

1.6 Technology stack

- Frontend :- Html , CSS, JavaScript
- Blockchain:- Ethereum, Solidity, Ganache
- Platform :- Visual Studio Code

1.7 Benefits for environment & Society

- Enhanced security: Your data is sensitive and crucial, and blockchain can significantly change how your critical information is viewed.

- Greater transparency: Without blockchain, each organization has to keep a separate database.

- Instant traceability: Blockchain creates an audit trail that documents the provenance of an asset at every step on its journey.

- Automation: Transactions can even be automated with “smart contracts,” which increase your efficiency and speed the process even further.

2. Project Design

2.1 Proposed System

1. Product Traceability:

- A product can be added to the platform with all the necessary details and can trace their owner at any time.

2. Security:

- Blockchain will be used to store all the data also QR code will prevent the data breaches.

3. Product Verification:

- Customer can verify the products using generated QR by just scanning it.

2.2 Design(Flow Of Modules)

Our product anti-counterfeiting system based on Blockchain is composed of three roles, The Manufacturer Role, The Seller Role, and The Consumer role.

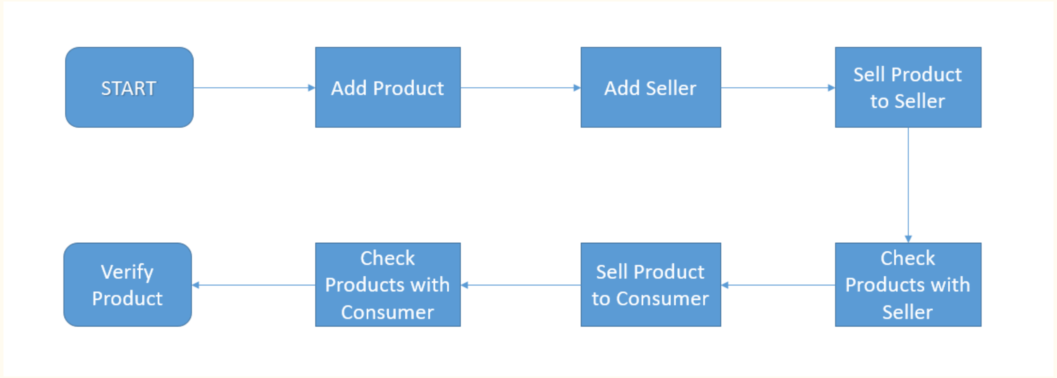


Fig. Workflow

- Manufacturer: Manufacturer logs into the manufacturer account and adds other required details of the product to add catalogue in the website. Manufacturer is the main entity who creates the blockchain and push the starting values into it. Functions like,
  i] Add Products, ii] Add Seller, iii] Sell Product to Seller, iv] Query Seller

- Seller: Supplier logs into supplier account and Seller is the intermediate identity who takes the authority from Manufacturer and sells the product to the consumer. Functions like,
  i] Show Products for sale, ii] Can sell products to consumer and generate QR code.

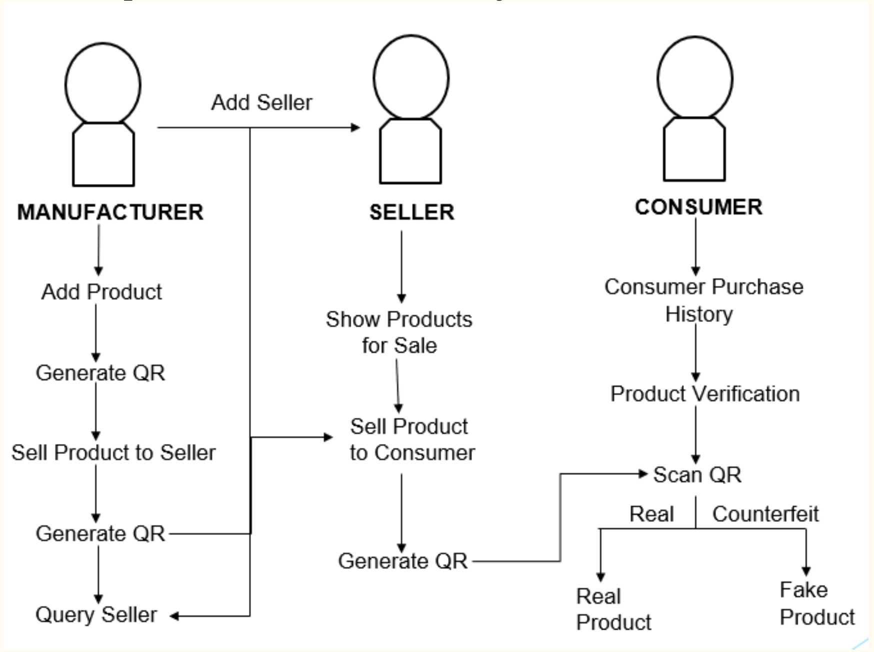
- Consumer: Customers can check the integrity of the product by scanning QR code which will list the history of transactions and thus verifying the genuinity of the product. Consumer is the end user here who will check the genuineness of product. Functions like,
  i] Consumer purchase history, ii] Product Verification.

2.3 Description Of Use Case

- Manufacturer will be the admin of the website. They can control the supply chain.
- The supplier will be the middle man who is providing that product their name and handling over to Consumer.
- End Consumer then gets the genuine product which follows this cycle.

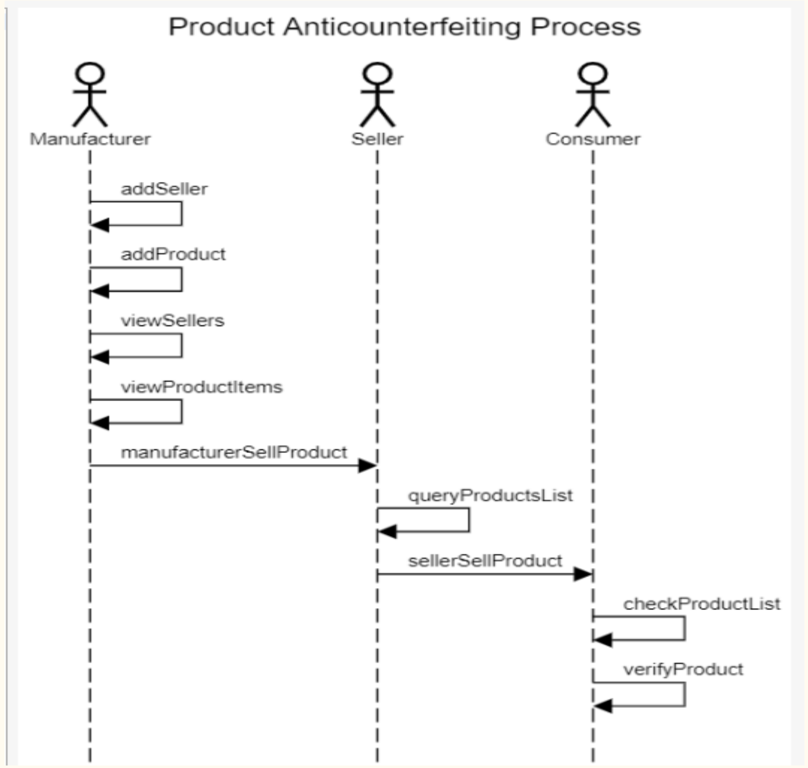
2.4 Activity diagram

The figure shown is the complete activity diagram of the proposed system. First the user will be assigned with the roles like Manufacture, Seller and Consumer. The home page would be displayed. By selecting the roles we can perform the different actions respective to the roles. Every module is connected with each other.



2.5 Sequence Diagram

The sequence diagram shows the entire sequence of flow of the data in the system. In the initial step the Manufacturer will add product and sellers with their details. Then only Manufacture can sell that added product to any particular Seller with the allotted seller code. Manufacturer can also check the added Seller details. Next, Seller can check Products for Sale and he can also sell that product to the Consumer. Finally, Consumer can check the products with him and can verify any product using QR code.

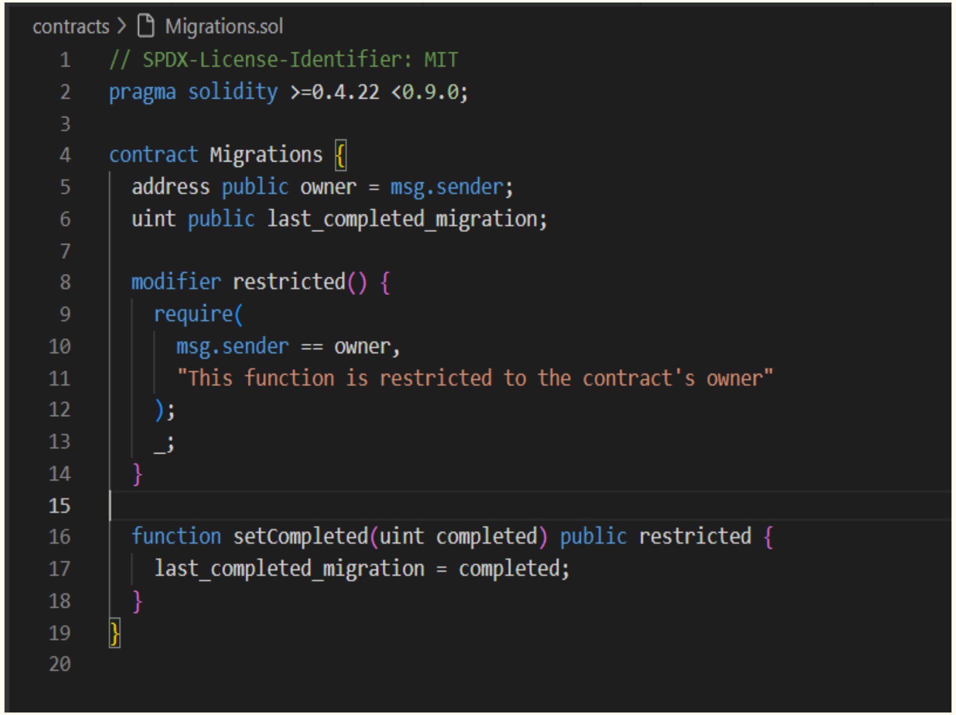


3. Implementation

3. Implementation

Project implementation consists of visions and plans with which we are supposed to build the end product. This includes the logical conclusion, after evaluating, deciding, visioning, planning and finding the other resources for the project. Technical implementation is one of the major aspects of executing a project.

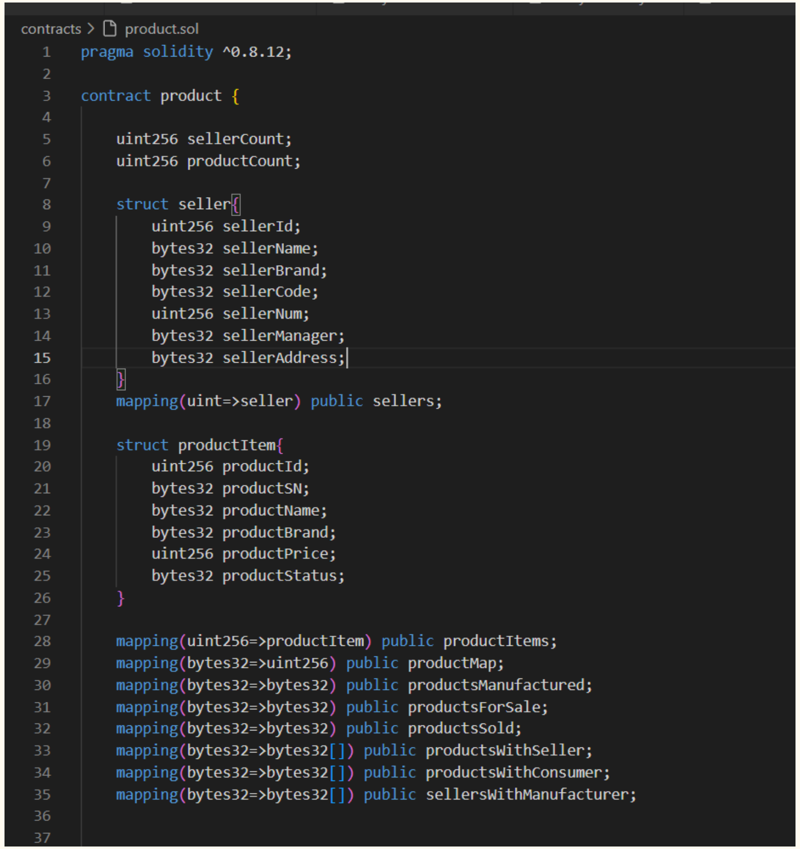
In fig. 3.1 code is implementing a smart contract in the Solidity programming language. The smart contract is called "Migrations".


contracts > Migrations.sol

```solidity
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.9.0;

contract Migrations {
  address public owner = msg.sender;
  uint public last_completed_migration;

  modifier restricted() {
    require(
      msg.sender == owner,
      "This function is restricted to the contract's owner"
    );
    _;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }
}
```

Figure 3.1 Migration Code

This code declares multiple mappings that are used to manage a product inventory system in Solidity.

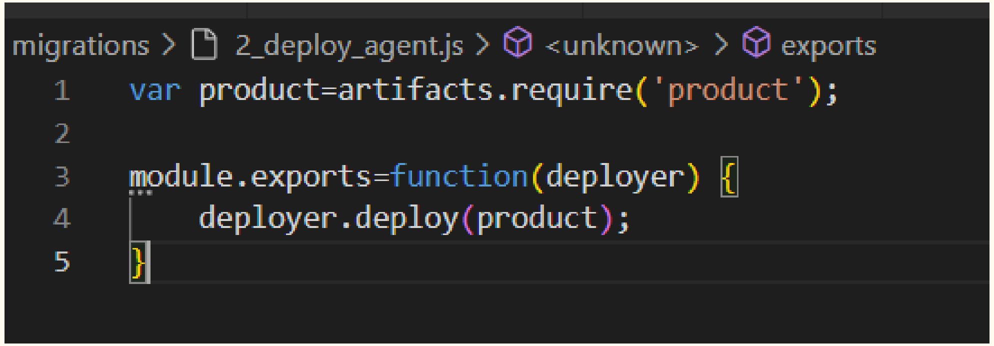


contracts > product.sol

```solidity
pragma solidity ^0.8.12;

contract product {

    uint256 sellerCount;
    uint256 productCount;

    struct seller{
        uint256 sellerId;
        bytes32 sellerName;
        bytes32 sellerBrand;
        bytes32 sellerCode;
        uint256 sellerNum;
        bytes32 sellerManager;
        bytes32 sellerAddress;
    }
    mapping(uint=>seller) public sellers;

    struct productItem{
        uint256 productId;
        bytes32 productSN;
        bytes32 productName;
        bytes32 productBrand;
        uint256 productPrice;
        bytes32 productStatus;
    }

    mapping(uint256=>productItem) public productItems;
    mapping(bytes32=>uint256) public productMap;
    mapping(bytes32=>bytes32) public productsManufactured;
    mapping(bytes32=>bytes32) public productsForSale;
    mapping(bytes32=>bytes32) public productsSold;
    mapping(bytes32=>bytes32[]) public productsWithSeller;
    mapping(bytes32=>bytes32[]) public productsWithConsumer;
    mapping(bytes32=>bytes32[]) public sellersWithManufacturer;


```

Figure 3.2 Smart Contract to manage Product

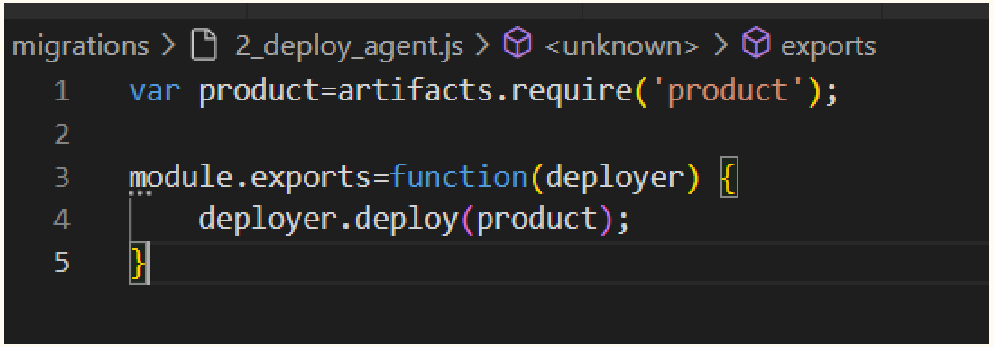
This code deploys the "Migrations" contract to the blockchain network using the Truffle framework, which allows developers to easily deploy, test, and interact with smart contracts.


migrations > 2_deploy_agent.js > <unknown> > exports

```
var product=artifacts.require('product');

module.exports=function(deployer) {
    deployer.deploy(product);
}
```

Figure 3.3 Deploying Smart Contracts

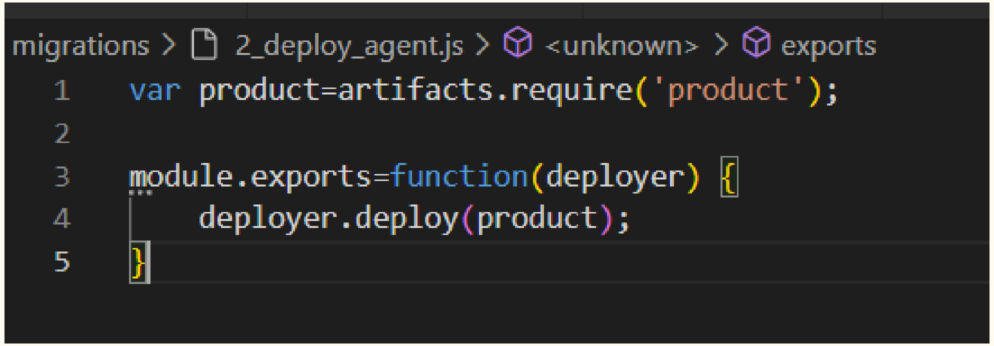
Code that uses the Truffle framework to deploy a smart contract called "product" to a blockchain network.


migrations > 2_deploy_agent.js > <unknown> > exports

```
var product=artifacts.require('product');

module.exports=function(deployer) {
    deployer.deploy(product);
}
```

Figure 3.4 Deploying Product Smart Contract on Blockchain

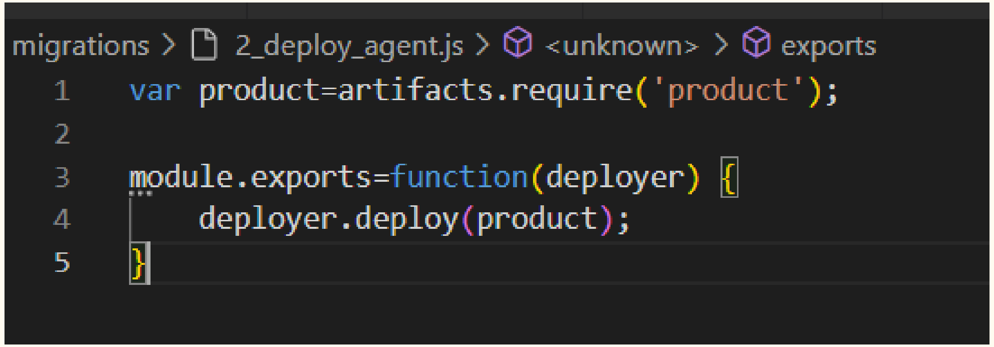
This code deploys the "Migrations" contract to the blockchain network using the Truffle framework, which allows developers to easily deploy, test, and interact with smart contracts.


migrations > 2_deploy_agent.js > <unknown> > exports

```
var product=artifacts.require('product');

module.exports=function(deployer) {
    deployer.deploy(product);
}
```

Figure 3.3 Deploying Smart Contracts

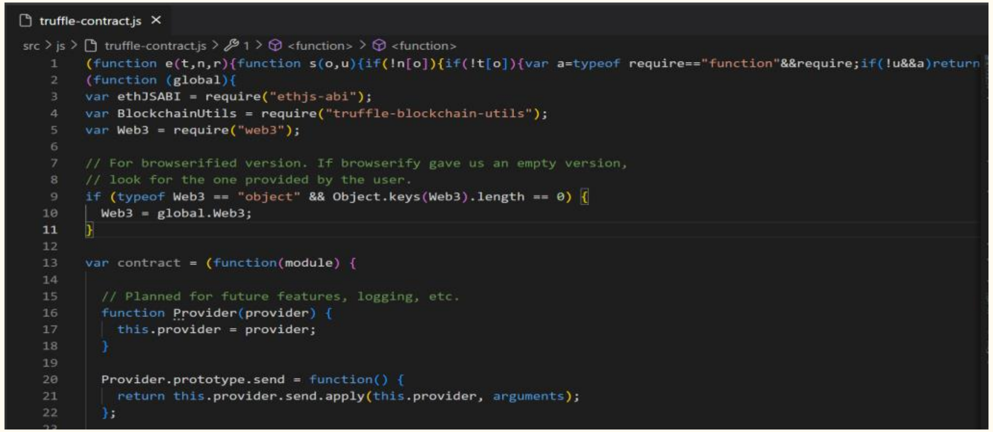
Code that uses the Truffle framework to deploy a smart contract called "product" to a blockchain network.


migrations > 2_deploy_agent.js > <unknown> > exports

```
var product=artifacts.require('product');

module.exports=function(deployer) {
    deployer.deploy(product);
}
```

Figure 3.4 Deploying Product Smart Contract on Blockchain

JavaScript code that defines a module for interacting with Ethereum smart contracts.



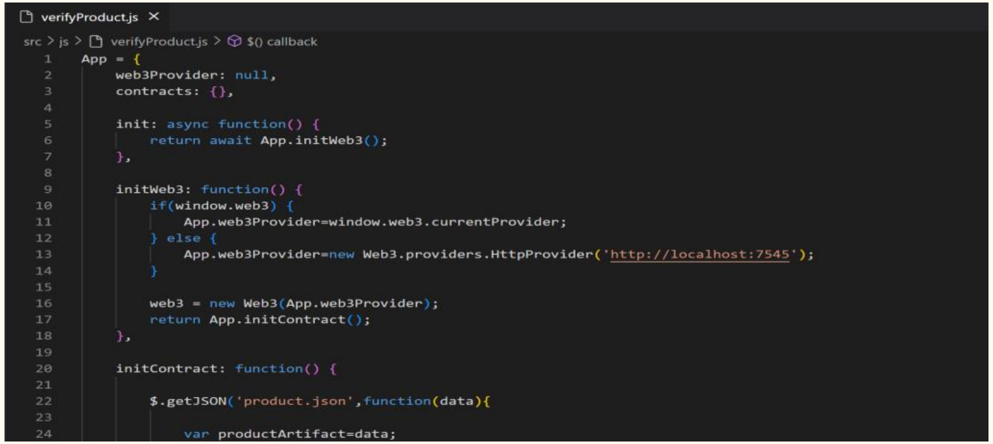
truffle-contract.js ×

src > js > truffle-contract.js > 1 > <function> > <function>

```javascript
(function e(t,n,r){function s(o,u){if(!n[o]){if(!t[o]){var a=typeof require=="function"&&require;if(!u&&a)return
(function (global){
var ethJSABI = require("ethjs-abi");
var BlockchainUtils = require("truffle-blockchain-utils");
var Web3 = require("web3");

// For browserified version. If browserify gave us an empty version,
// look for the one provided by the user.
if (typeof Web3 == "object" && Object.keys(Web3).length == 0) {
  Web3 = global.Web3;
}

var contract = (function(module) {

  // Planned for future features, logging, etc.
  function Provider(provider) {
    this.provider = provider;
  }

  Provider.prototype.send = function() {
    return this.provider.send.apply(this.provider, arguments);
  };
```

Figure 3.5 Interacting with Smart Contracts

A JavaScript code that defines an object App with properties and methods for interacting with a smart contract.



verifyProduct.js ×

src > js > verifyProduct.js > $() callback

```javascript
App = {
    web3Provider: null,
    contracts: {},

    init: async function() {
        return await App.initWeb3();
    },

    initWeb3: function() {
        if(window.web3) {
            App.web3Provider=window.web3.currentProvider;
        } else {
            App.web3Provider=new Web3.providers.HttpProvider('http://localhost:7545');
        }

        web3 = new Web3(App.web3Provider);
        return App.initContract();
    },

    initContract: function() {

        $.getJSON('product.json',function(data){

            var productArtifact=data;
```

Figure 3.6 Properties and Methods for Interacting with a Smart Contract

Here, is the home page of the E-commerce website as shown in Fig 3.7

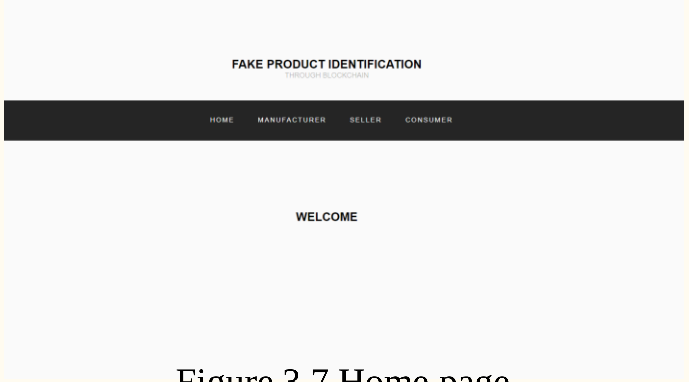
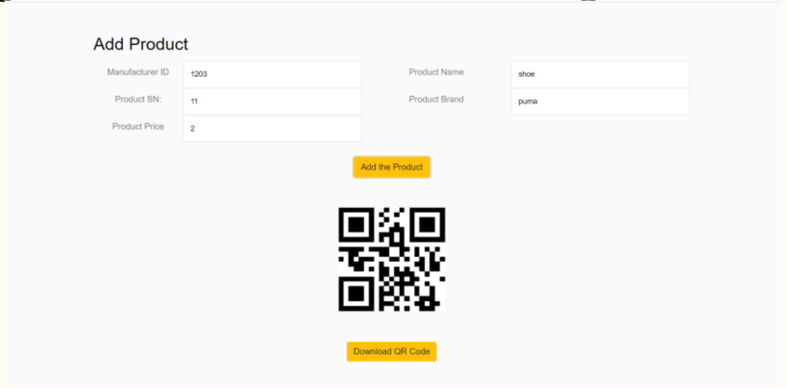


Figure 3.7 Home page

This is the add product page where Manufacturer can add products to the database with all the required details and it will generate a QR code as shown in fig. 3.8

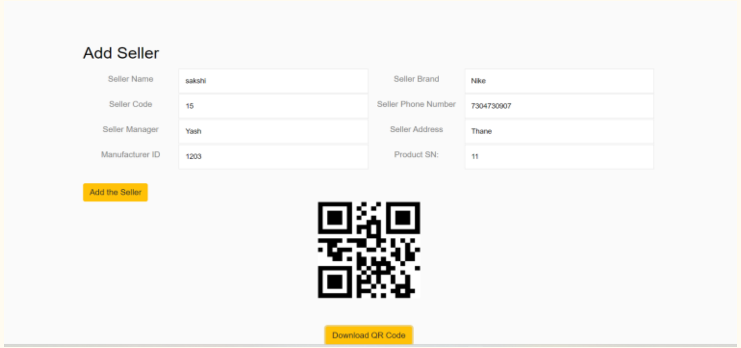


Add Product

Manufacturer ID	1203	Product Name	shoe
Product SN:	11	Product Brand	puma
Product Price	2		

Figure 3.8 Add Product Page

This is the add seller page where Manufacturer can add seller to the database with all the required details and it will generate a QR code as shown in fig. 3.9

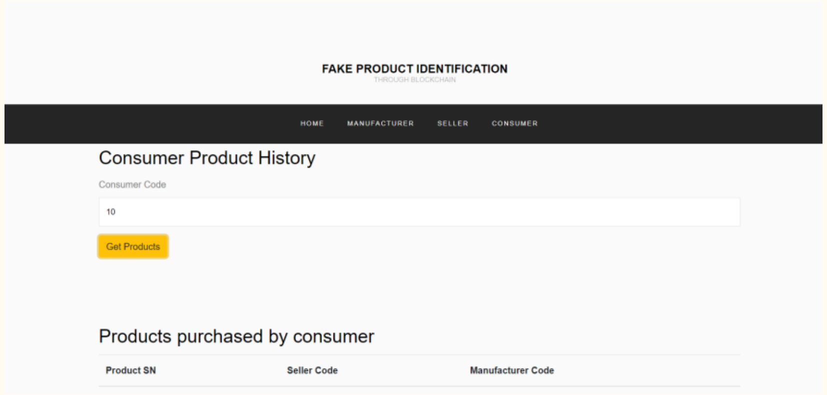


Add Seller

Seller Name	sakshi	Seller Brand	Nike
Seller Code	15	Seller Phone Number	7304730907
Seller Manager	Yash	Seller Address	Thane
Manufacturer ID	1203	Product SN:	11

Figure 3.9 Add Seller Page

This is the consumer product history page, where consumer can submit their code and get the list of products bought by them.



Consumer Product History

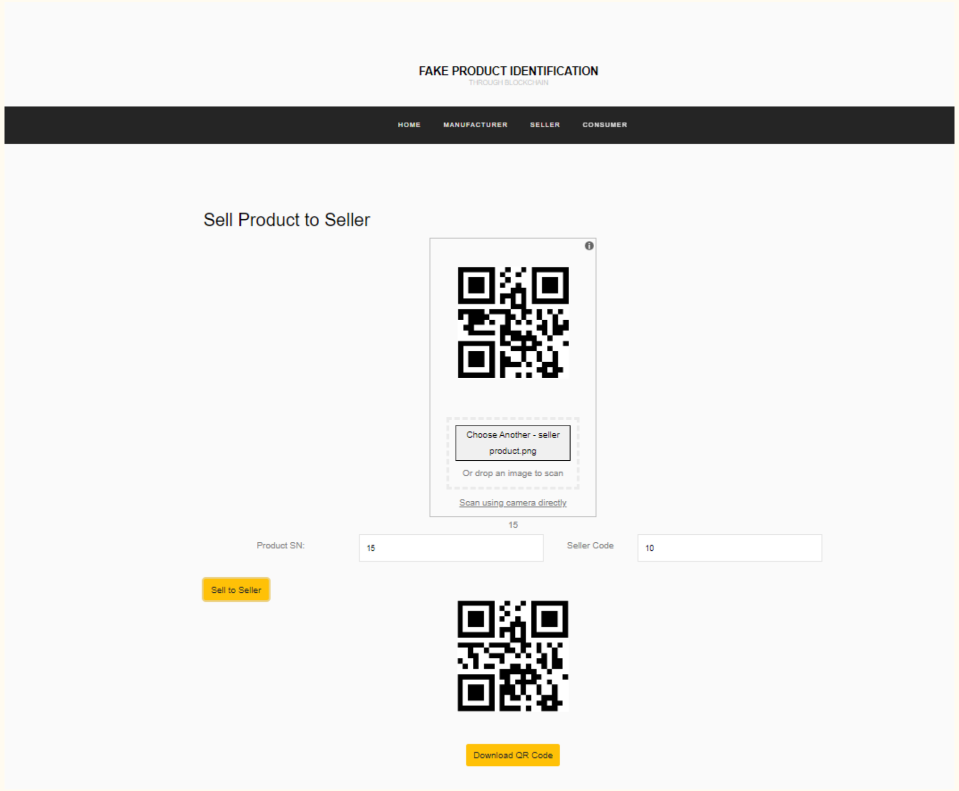
Consumer Code: 10

Products purchased by consumer

Product SN	Seller Code	Manufacturer Code

Figure 3.10 Consumer Product History Page

This is the sell product to seller page, where Manufacturer can add QR code generated in previous step and seller code to whom the product authority should be given.



Sell Product to Seller

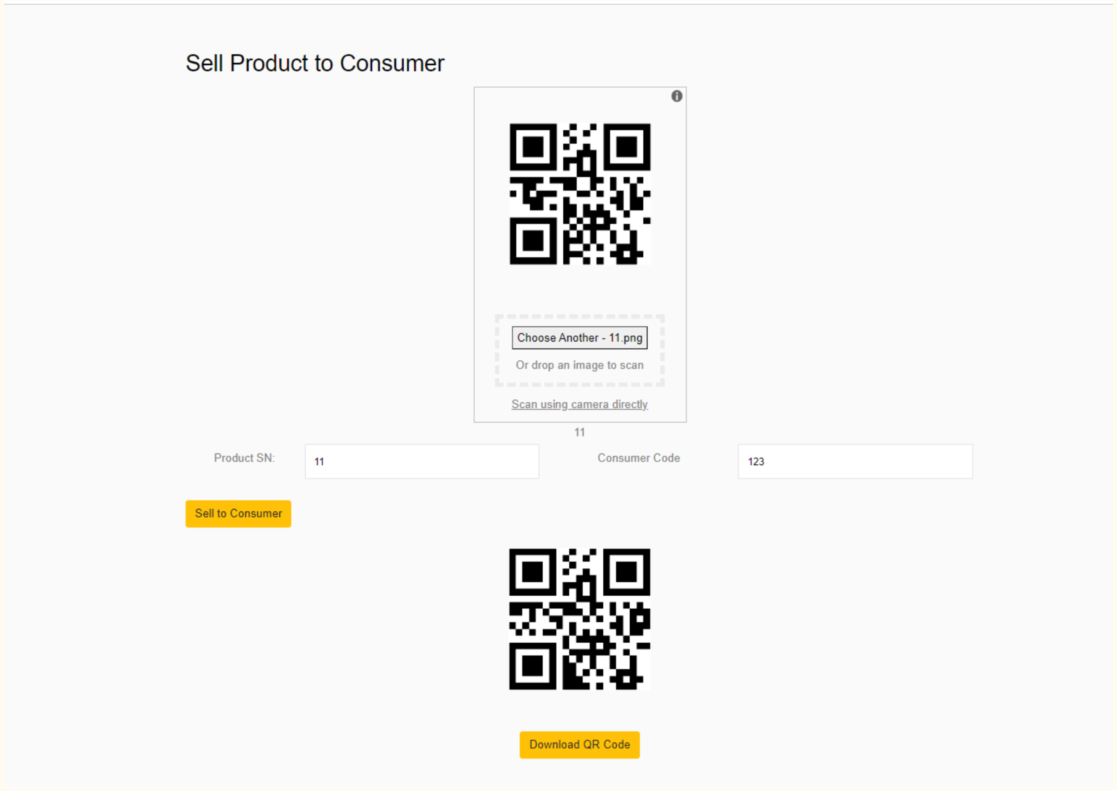
15

Product SN: 15

Seller Code 10

Figure 3.11 Sell Product to Seller Page

This is the sell product to consumer page, where Seller can add QR code generated in previous step and consumer code to whom the product authority should be given.



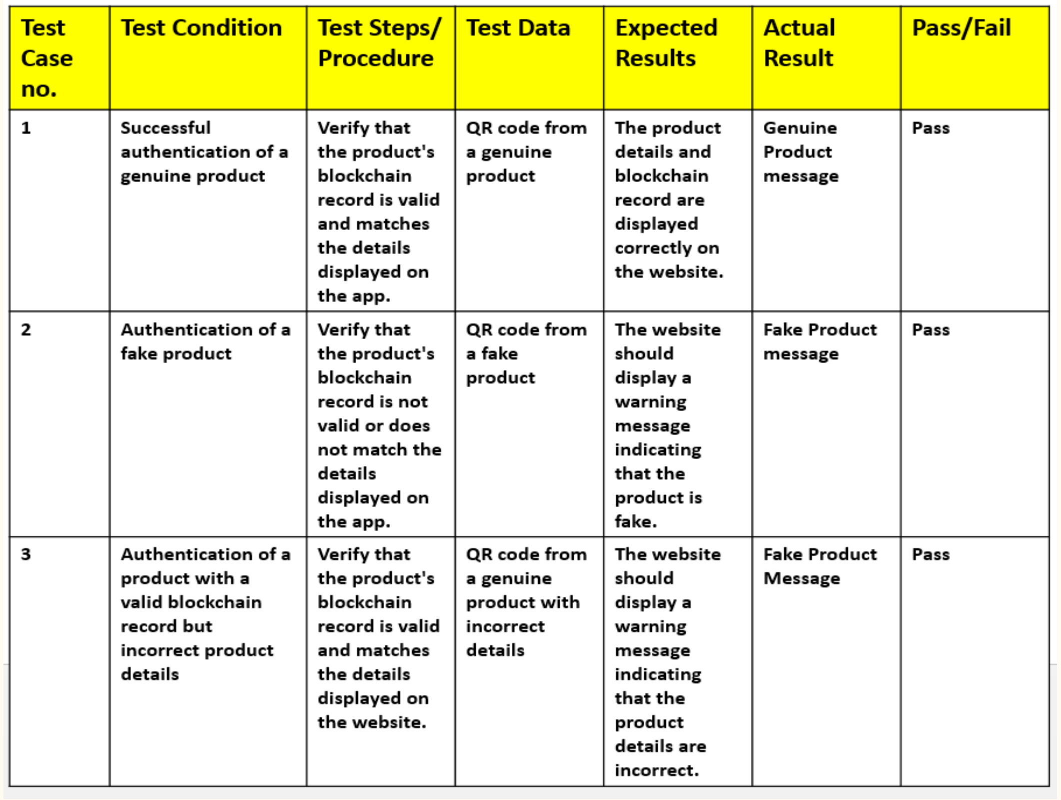
Product SN: 11

Consumer Code: 123

Figure 3.12 Sell Product to Consumer Page

4. Testing

4.1 Unit Testing



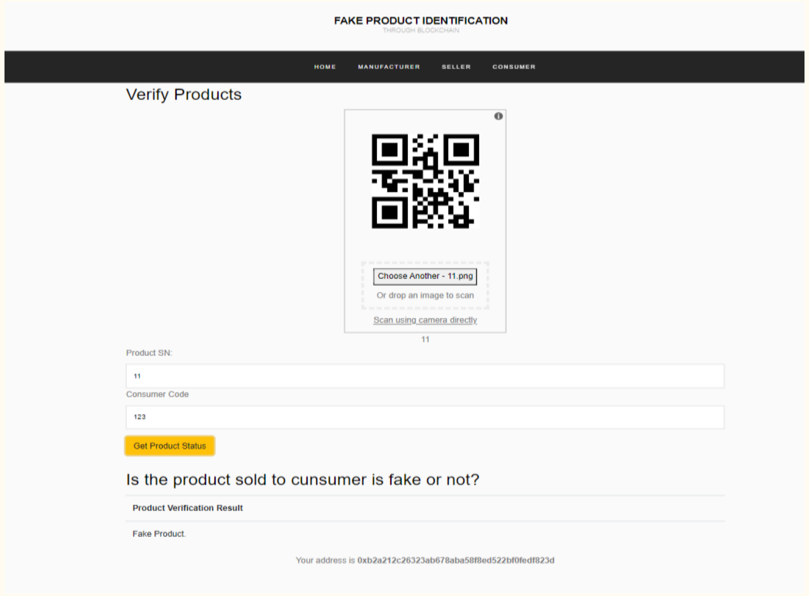
Test Case no.	Test Condition	Test Steps/ Procedure	Test Data	Expected Results	Actual Result	Pass/Fail
1	Successful authentication of a genuine product	Verify that the product's blockchain record is valid and matches the details displayed on the app.	QR code from a genuine product	The product details and blockchain record are displayed correctly on the website.	Genuine Product message	Pass
2	Authentication of a fake product	Verify that the product's blockchain record is not valid or does not match the details displayed on the app.	QR code from a fake product	The website should display a warning message indicating that the product is fake.	Fake Product message	Pass
3	Authentication of a product with a valid blockchain record but incorrect product details	Verify that the product's blockchain record is valid and matches the details displayed on the website.	QR code from a genuine product with incorrect details	The website should display a warning message indicating that the product details are incorrect.	Fake Product Message	Pass

Fig. Testcases

5. Result

5.1 Verify Product

QR codes plays a very important role in determining Genuine and Fake products. QR code generated in the last step should be submitted to fetch the product id and the by providing Consumer code we can check that is the product sold to consumer is fake or not.



Verify Products

11

Product SN: 11

Consumer Code: 123

Is the product sold to cunsumer is fake or not?

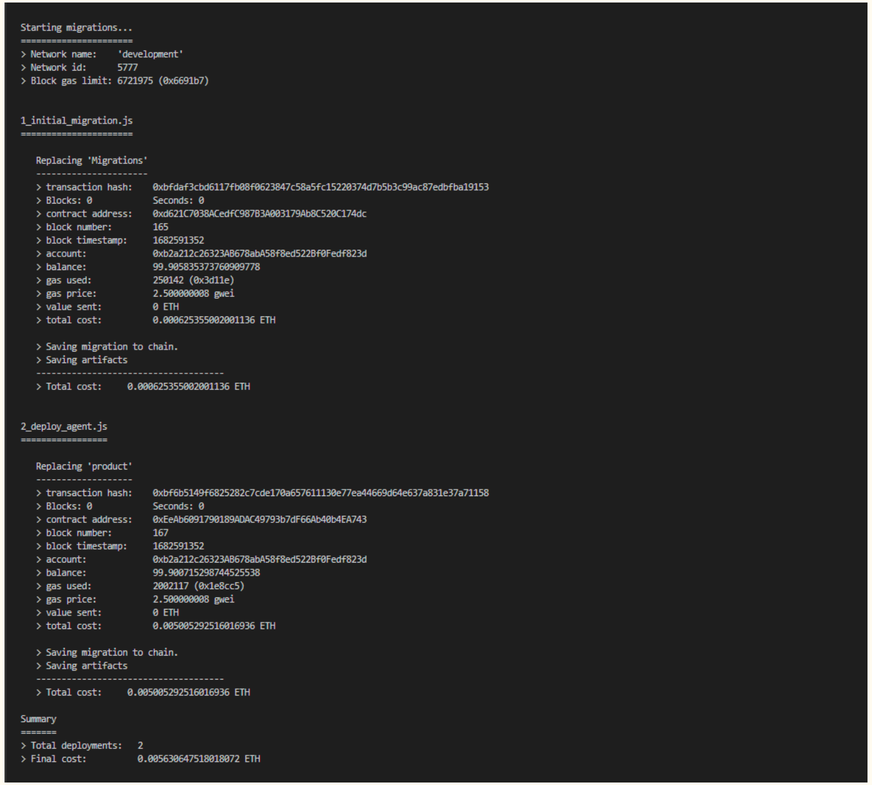
Product Verification Result

Fake Product.

Your address is 0xb2a212c26323ab678aba58f8ed522bf0fedf823d

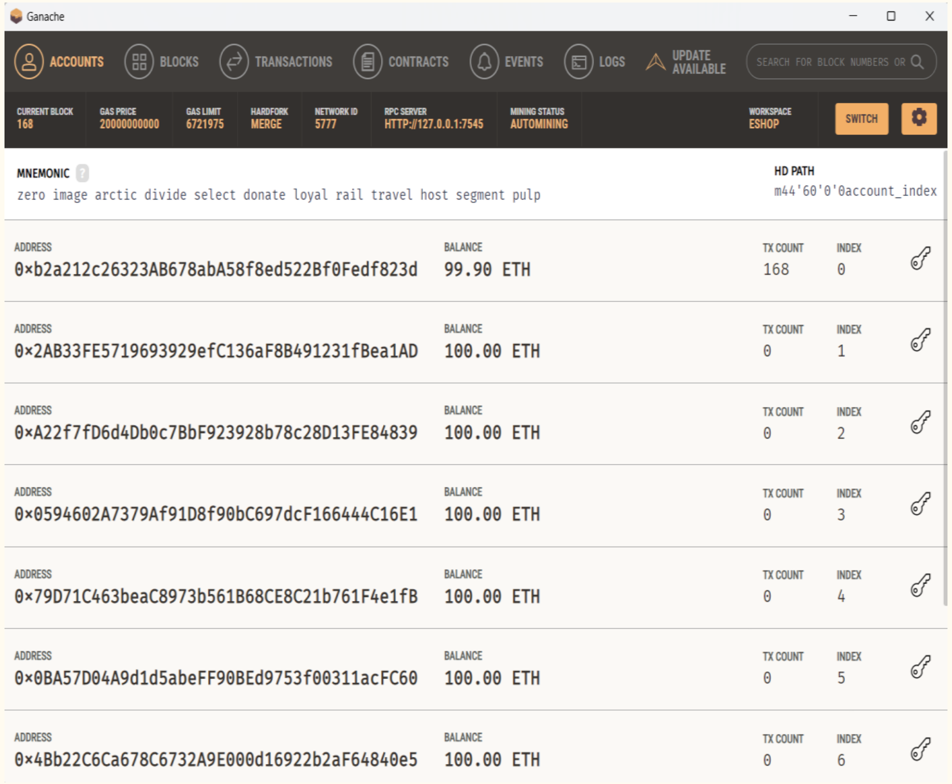
Fig. Verify Product Page

5.2 Smart Contracts and Blocks



```
Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975 (0x6691b7)


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0xbfdaf3cbd6117fb08f0623847c58a5fc15220374d7b5b3c99ac87edbfba19153
   > Blocks: 0            Seconds: 0
   > contract address:    0xd621C7038ACedfC987B3A003179Ab8C520C174dc
   > block number:        165
   > block timestamp:     1682591352
   > account:             0xb2a212c26323AB678abA58f8ed522Bf0Fedf823d
   > balance:             99.905835373760909778
   > gas used:            250142 (0x3d11e)
   > gas price:           2.500000008 gwei
   > value sent:          0 ETH
   > total cost:          0.000625355002001136 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:     0.000625355002001136 ETH


2_deploy_agent.js
=================

   Replacing 'product'
   -------------------
   > transaction hash:    0xbf6b5149f6825282c7cde170a657611130e77ea44669d64e637a831e37a71158
   > Blocks: 0            Seconds: 0
   > contract address:    0xEeAb6091790189ADAC49793b7dF66Ab40b4EA743
   > block number:        167
   > block timestamp:     1682591352
   > account:             0xb2a212c26323AB678abA58f8ed522Bf0Fedf823d
   > balance:             99.900715298744525538
   > gas used:            2002117 (0x1e8cc5)
   > gas price:           2.500000008 gwei
   > value sent:          0 ETH
   > total cost:          0.005005292516016936 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:     0.005005292516016936 ETH

Summary
=======
> Total deployments:   2
> Final cost:          0.005630647518018072 ETH
```

Fig. Deploying Smart Contract on the Blockchain Network



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
168	20000000000	6721975	MERGE	5777	HTTP://127.0.0.1:7545	AUTOMINING	ESHOP

MNEMONIC
zero image arctic divide select donate loyal rail travel host segment pulp

HD PATH
m44'60'0'0account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×b2a212c26323AB678abA58f8ed522Bf0Fedf823d	99.90 ETH	168	0
0×2AB33FE5719693929efC136aF8B491231fBea1AD	100.00 ETH	0	1
0×A22f7fD6d4Db0c7BbF923928b78c28D13FE84839	100.00 ETH	0	2
0×0594602A7379Af91D8f90bC697dcF166444C16E1	100.00 ETH	0	3
0×79D71C463beaC8973b561B68CE8C21b761F4e1fB	100.00 ETH	0	4
0×0BA57D04A9d1d5abeFF90BEd9753f00311acFC60	100.00 ETH	0	5
0×4Bb22C6Ca678C6732A9E000d16922b2aF64840e5	100.00 ETH	0	6

Fig. Records Uploaded on local Blockchain

6. Conclusion and Future Scope

6.1 Conclusion and Future Scope

- The system provides a secure and efficient framework for product authentification.
- Useful for the Seller as well as Consumer to maintain a transparent and secure cycle of transaction.
- Promising future with the potential to revolutionize the way we verify the authenticity of products in online transactions.
- It will be useful to make a change in the society by adopting a new trend on E-commerce shopping platforms.

References

- [1] B.Prabhu Shankar, Dr. R.Jayavadivel, "A Survey Of Counterfeit Product Detection", International Journal of Scientific and Technology Research Volume 8, 12, December 2019.
- [2] Si Chen; Rui Shi; Zhuangyu Ren; Jiaqi Yan; Yani Shi; Jinyu Zhang, "A Blockchain-Based Supply Chain Quality Management Framework", 2017 IEEE 14th International Conference on E-Business Engineering (ICEBE), 04-06 November 2019.
- [3] Jinhua Ma; Shih-Ya Lin; Xin Chen; Hung-Min Sun; Yeh-Cheng Chen; Huaxiong Wang, "A Blockchain-Based Application System for Product Anti-Counterfeiting", IEEE Access (Volume: 8), 06 February 2020.

Paper Publication

Paper entitled “Eshop: Comprehensive Blockchain based Web Framework for E-Commerce” is submitted at “7th Edition Of ITC (IEEE International Test Conference of India 2023)” by “Sakshi Shinde, Anurag Singh, Shubhangi Lanke and Prof. Manjusha Kashilkar.”

Thank You