

Fake Product Identification System Using Blockchain Technology

Introduction to Blockchain

A blockchain is a digital and distributed ledger of transactions, recorded and replicated in real-time across a network of computers or nodes. Every transaction must be cryptographically validated via a consensus mechanism executed by the nodes before being permanently added as a new "block" at the end of the "chain." There is no need for a central authority to approve the transaction, which is why blockchain is sometimes referred to as a peer-to-peer trustless mechanism.

Blockchain can be thought of as a linked list with each node containing multiple transactions. Each transaction has a hash that depends on the previous transactions hash as well. So we can see that the order of transactions is important. If we were to change one transaction somewhere, it would have a ripple effect and change the hash of all subsequent transactions. This is one of the reasons why blockchain is a powerful medium for storing transactions.

The placing of a transaction in a block is called a successful conclusion to a proof of work challenge, and is carried out by special nodes called miners. Proof of Work is a system that requires some work from the service requester, usually meaning processing time by a computer. Producing a proof of work is a random process with low probability, so normally a lot of trial and error is required for a valid proof of work to be generated. When it comes to Bitcoins, hash is what serves as a proof of work. Miners on a Blockchain are nodes that produce blocks by solving proof of work problems. If a miner produces a block that is approved by an electronic consensus of nodes then the miner is rewarded with coins. This essentially is the crux of blockchain. Proof of Work is what is keeping all transactions on the blockchain secure and protecting it from malicious attempts to alter these transactions.

Problem Statement

Every popular brand has fake manufacturers selling a counterfeit item with misleading and invalid labels, which are sold at cheaper rates. Even the company experts may not be able to distinguish between counterfeit and original items.

Suppose we come across a counterfeit item - we need to be able to identify that it is indeed fake through its QR code.

Our Solution to the Problem

We can add users and nodes into the network at any time. There is a single blockchain that stores the transactions. Each blockchain has several blocks and each transaction has several transactions. Each transaction consists of the following details:

- Date and time
- Sender/Seller name and ID
- Recipient name and ID
- Product name and ID
- Price

In case the seller enters a wrong product code (i.e., a counterfeit item), the transaction is verified using the [Zero Knowledge Proof](#) by verifying whether the item is original or not. If the transaction is found to be invalid, the seller is added to the list of suspicious users and the transaction is not added to the blockchain.

Zero-Knowledge Proof

In cryptography, a zero-knowledge proof or zero-knowledge protocol is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true while the prover avoids conveying any additional information apart from the fact that the statement is indeed true. The essence of zero-knowledge proofs is that it is trivial to prove that one possesses knowledge of certain information by simply revealing it; the challenge is to prove such possession without revealing the information itself or any additional information.

If proving a statement requires that the prover possess some secret information, then the verifier will not be able to prove the statement to anyone else without possessing the secret information. The statement being proved must include the assertion that the prover has such knowledge, but without including or transmitting the knowledge itself in the assertion. Otherwise, the statement would not be proved in zero-knowledge because it provides the verifier with additional information about the statement by the end of the protocol. A zero-knowledge proof of knowledge is a special case when the statement consists only of the fact that the prover possesses the secret information.

Zero-Knowledge Proof Algorithm

Alice has sensitive data x for which she chooses two numbers p and g . p can be a large prime and g is a generator for p . She calculates y as $y = g^x \mod p$. Now she performs the following steps to create a zero knowledge proof for x .

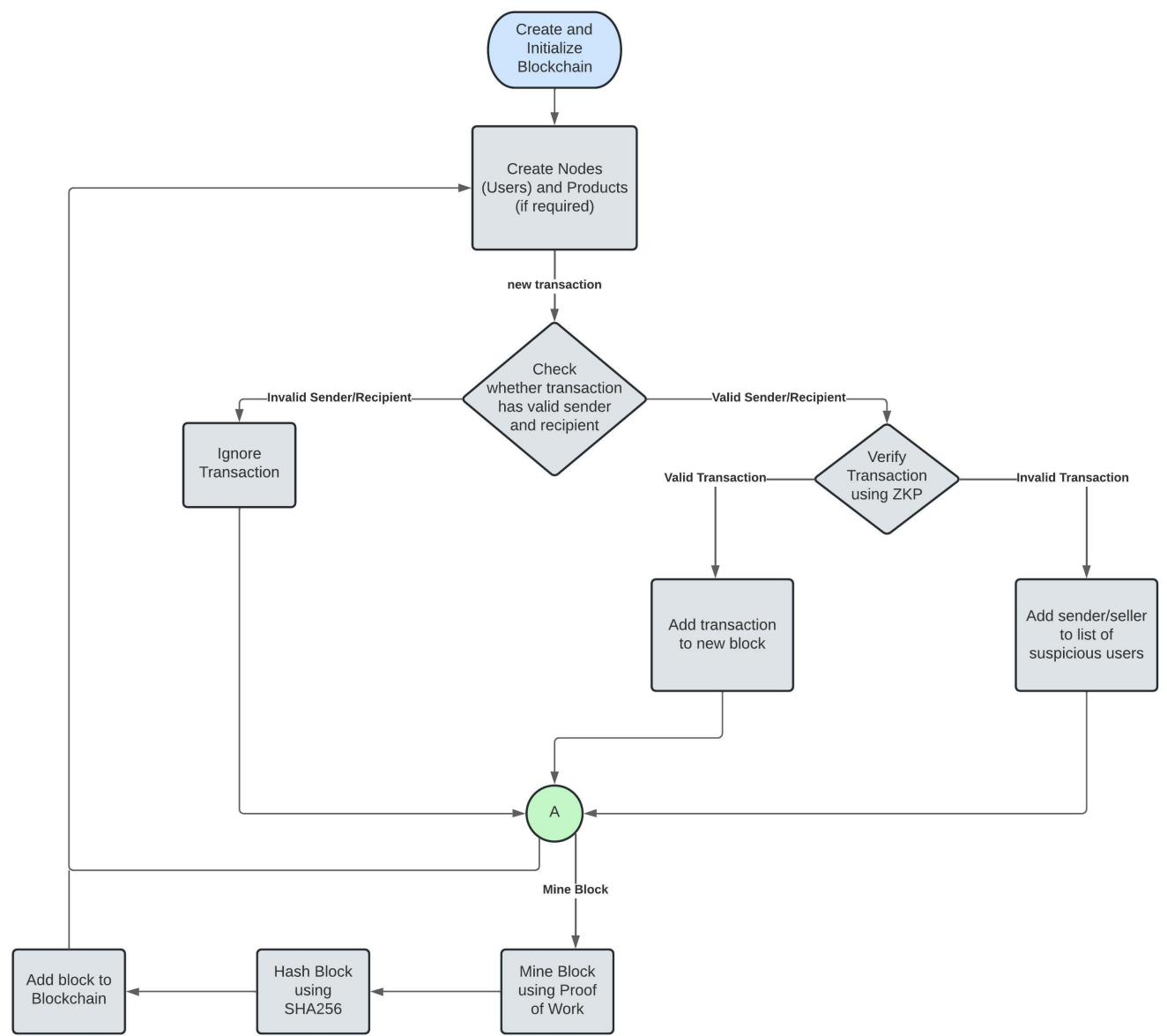
1. Alice chooses a random number $r < p - 1$ and sends it to Bob as $h = g^r \mod p$
2. Bob receives h and sends back a random bit b (0/1).
3. Alice sends $s = (r + bx) \mod (p - 1)$ to Bob.
4. Bob computes $g^s \mod p$ which should equal $hy^b \mod p$

Here Bob acts as a verifier and checks if Alice knows the value of x without actually getting to know what x is.

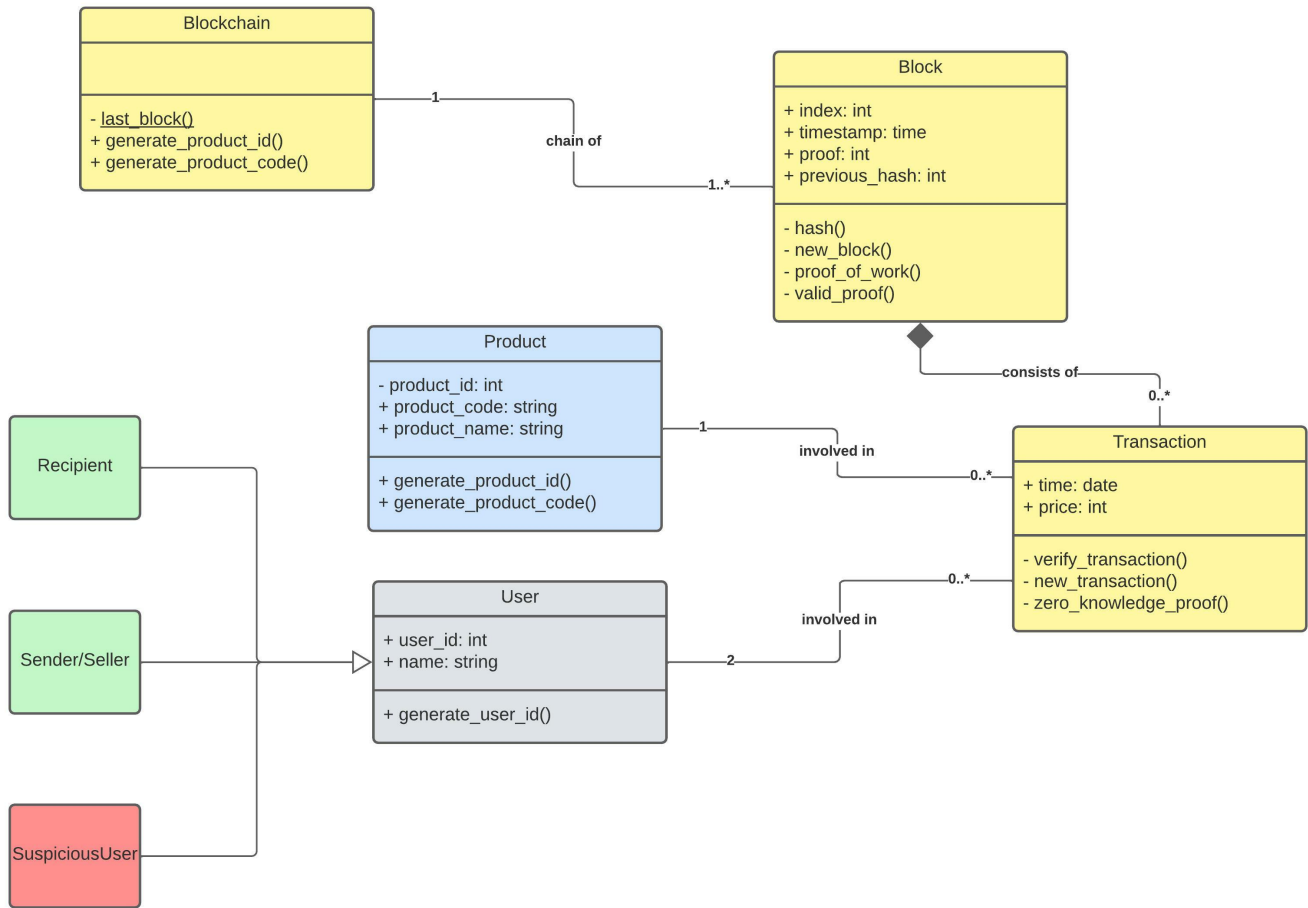
Implementation of Zero-Knowledge Proof

We have used Python's default `pow()` for modular exponentiation, and the `random` library to generate the random bit.

Flowchart depicting the control flow of the Blockchain

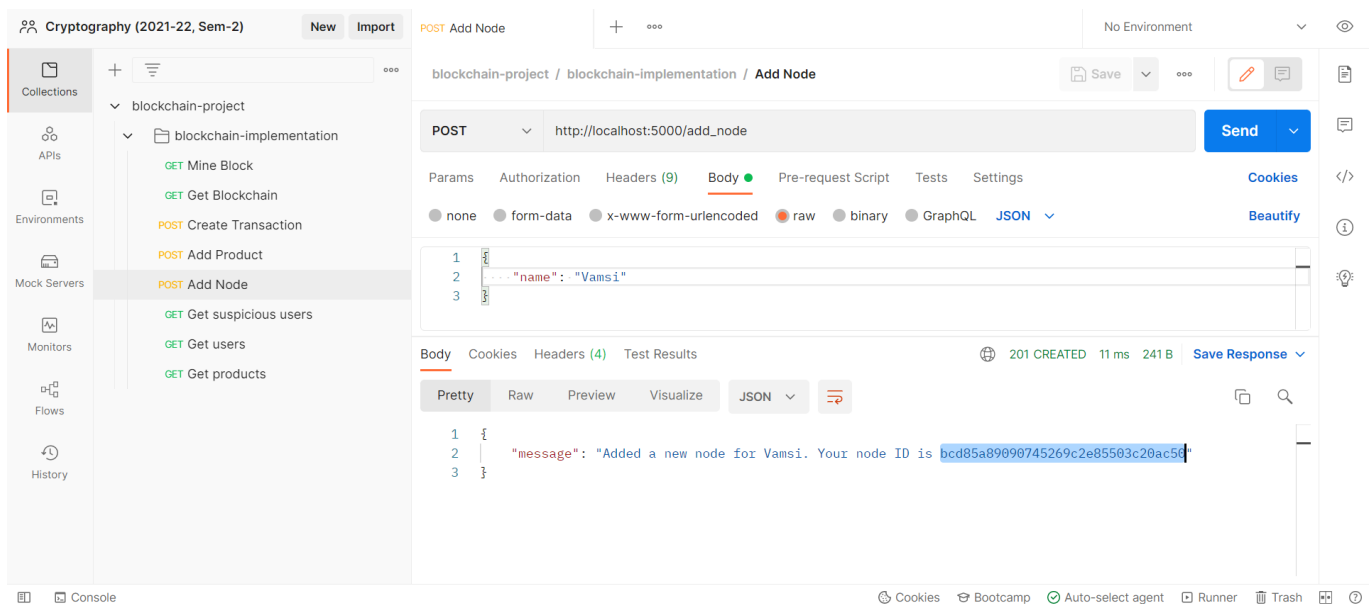


UML Diagram of classes used



Screenshots of the working Application

Add Node (User) to the Blockchain



Add a Product to a Blockchain

Cryptography (2021-22, Sem-2)

NewImport

POST Add Node

GET Get users

POST Add Product

No Environment

blockchain-project / blockchain-implementation / Add Product

Save

POST

http://localhost:5000/add_product

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

product_name": "laptop"

Body

Cookies

Headers (4)

Test Results

Status: 201 CREATED

Time: 86 ms

Size: 277 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

"message": "Added a new product laptop. The code for this product is Ly8nESkbz2kQfoVoKslc0yJlGwD5nMyY. Keep a note of this."

Console

Cookies

Bootcamp

Auto-select agent

Runner

Trash

Show all users of the Blockchain

Cryptography (2021-22, Sem-2)

NewImport

POST Add Node

GET Get users

No Environment

blockchain-project / blockchain-implementation / Get users

Save

GET

http://localhost:5000/users

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 8 ms

Size: 332 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

"users": [

{

"ID": "bcd85a89898745269c2e85583c28ac58",

"Name": "Vamsi"

},

{

"ID": "f4316421ae5b48768f91a5f42f6ae24d",

"Name": "Ruban"

},

{

"ID": "6283c51acc4842d8acab799cb487128f",

"Name": "Subienay"

}

]

Console

Cookies

Bootcamp

Auto-select agent

Runner

Trash

Show all products in the Blockchain

Cryptography (2021-22, Sem-2)

NewImport

POST Add NodeGET Get usersPOST Add ProductGET Get products

+...No Environment

blockchain-project / blockchain-implementation / Get products

Save

Send

GET

http://localhost:5000/products

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettings

Cookies

BodyCookiesHeaders (4)Test Results

Status: 200 OKTime: 7 msSize: 469 BSave Response

PrettyRawPreviewVisualizeJSON

12345678910111213141516171819

```
1  {
2    "products": [
3      {
4        "product_code": "Ly8nESkbz2kQfoYoKslc0yJIgW05nMyY",
5        "product_name": "laptop"
6      },
7      {
8        "product_code": "cHf5UvdQxsDTdxoyaYtFtR7iszVKbFxfj",
9        "product_name": "smartphone"
10     },
11     {
12       "product_code": "YzonEEzfKwPELiB9Ici3IZXXV0z90EC4",
13       "product_name": "speaker"
14     },
15     {
16       "product_code": "eWTA0G5vSw1i13xUbmhSpDhSr3pD0Avv",
17       "product_name": "watch"
18     }
19   ]
20 }
```

Console

CookiesBootcampAuto-select agentRunnerTrash

Create a Transaction

Cryptography (2021-22, Sem-2)

NewImport

GET Get usersPOST Create TransactionGET Get products

+...No Environment

blockchain-project / blockchain-implementation / Create Transaction

Save

Send

POST

http://localhost:5000/transactions/new

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsSettings

Cookies

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

Beautiful

123456

```
1  {
2    "sender_id": "bcd85a89090745269c2e05503c20ac50",
3    "recipient_id": "f4316421ae5b40768f91a5f42f6ae24d",
4    "product_code": "Ly8nESkbz2kQfoYoKslc0yJIgW05nMyY",
5    "price": 60000
6  }
```

BodyCookiesHeaders (4)Test Results

Status: 201 CREATEDTime: 251 msSize: 201 BSave Response

PrettyRawPreviewVisualizeJSON

123

```
1  {
2    "message": "Transaction will be added to Block 2"
3  }
```

Console

CookiesBootcampAuto-select agentRunnerTrash

Mine a block

Cryptography (2021-22, Sem-2)

NewImport

Collections

+

blockchain-project

blockchain-implementation

GET Mine Block

GET Get Blockchain

POST Create Transaction

POST Add Product

POST Add Node

GET Get suspicious users

GET Get users

GET Get products

GET Get users

GET Get products

POST Create Transaction

GET Mine Block

+

...

No Environment

blockchain-project / blockchain-implementation / Mine Block

GET

http://localhost:5000/mine

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Body

Cookies

Headers (4)

Test Results

Status: 200 OK Time: 231 ms Size: 904 B Save Response

Pretty

Raw

Preview

Visualize

JSON

Send

```
1 {
2   "index": 2,
3   "message": "New Block Forged",
4   "previous_hash": "be84c07d99a1ad3a16522b9149040fe7ee2bab97733399fcbab7b342d03473c22",
5   "proof": 35293,
6   "transactions": [
7     {
8       "price": 60000,
9       "product": {
10        "product_code": "Ly8nESkbz2KfoYoKslc0yJigwD5nMyY",
11        "product_name": "laptop"
12      },
13      "recipient": {
14        "recipient_id": "f4316421ae5b40768f91a5f42f6ae24d",
15        "recipient_name": "Ruban"
16      },
17      "sender": {
18        "sender_id": "bcd85a89090745269c2e85503c20ac50",
19        "sender_name": "Vamsi"
20      }
21    }
22  ]
23 }
```

Console

Cookies Bootcamp Auto-select agent Runner Trash

```
1  {
2    "index": 2,
3    "message": "New Block Forged",
4    "previous_hash": "be84c07d99a1ad3a16522b9149040fe7ee2bab97733399fcba7b342d03473c22",
5    "proof": 35293,
6    "transactions": [
7      {
8        "price": 60000,
9        "product": {
10          "product_code": "Ly8nESkbz2kQfoYoKslc0yJIgWD5nMyY",
11          "product_name": "laptop"
12        },
13        "recipient": {
14          "recipient_id": "f4316421ae5b40768f91a5f42f6ae24d",
15          "recipient_name": "Ruban"
16        },
17        "sender": {
18          "sender_id": "bcd85a89090745269c2e85503c20ac50",
19          "sender_name": "Vamsi"
20        },
21        "time": "25/04/2022 21:57:52"
22      },
23      {
24        "price": 4000,
25        "product": {
26          "product_code": "YzonEEzFkwPELiB9Ici3IZXXVDz90EC4",
27          "product_name": "speaker"
28        },
29        "recipient": {
30          "recipient_id": "6203c51acc4042d8acab799cb487120f",
31          "recipient_name": "Subienay"
32        },
33        "sender": {
34          "sender_id": "f4316421ae5b40768f91a5f42f6ae24d",
35          "sender_name": "Ruban"
36        },
37        "time": "25/04/2022 22:00:17"
38      }
39    ]
40  }
```


Show Blockchain

Cryptography (2021-22, Sem-2)

NewImport

GET Get usersGET Get productsPOST Create TransactioGET Mine BlockGET Get Blockchain

+...No Environment

blockchain-project / blockchain-implementation / Get Blockchain

GEThttp://localhost:5000/chainSend

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTestsSettingsCookies

BodyCookiesHeaders (4)Test ResultsStatus: 200 OKTime: 5 msSize: 1.44 KBSave Response

PrettyRawPreviewVisualizeJSON

```
1{"chain": [
2  {
3    "index": 1,
4    "previous_hash": 1,
5    "proof": 100,
6    "timestamp": 1650903569.9812598,
7    "transactions": []
8  },
9  {
10   "index": 2,
11   "previous_hash": "be84c07d99a1ad3a16522b9149040fe7ee2bab97733399fcbab7b342d03473c22",
12   "proof": 35293,
13   "timestamp": 1650904226.1055238,
14   "transactions": [
15     {
16       "price": 60000,
17       "product": {
18         "product_code": "Ly8nESkbz2kQfoYoKslc0yJlgW05nMyY",
19         "product_name": "laptop"
20       },
21       "recipient": {
22         "recipient_id": "f4316421ae5b40768f91a5f42f6ae24d",
23         "recipient_name": "Ruban"
24       },
25       "sender": {
26         "sender_id": "bcd85a89090745269c2e85503c20ac50",
27         "sender_name": "Vamsi"
28       },
29       "time": "25/04/2022 21:57:52"
30     },
31     {
32       "price": 4000,
33       "product": {
34         "product_code": "YzonEEzFkwPELiB9Ici3IZXXV0z90EC4",
35         "product_name": "speaker"
36       },
37       "recipient": {
38         "recipient_id": "6203c51acc4042d8acab799cb487120f",
39         "recipient_name": "Subienay"
40       },
41       "sender": {
42         "sender_id": "f4316421ae5b40768f91a5f42f6ae24d",
43         "sender_name": "Ruban"
44       },
45       "time": "25/04/2022 22:00:17"
46     }
47   ],
48   "proof": 35089,
49   "timestamp": 1650904495.0981493,
50   "transactions": [
51     {
52       "price": 6000,
53       "product": {
54         "product_code": "eWTAOG5vSw1i13xUbmhSpDhSr3pD0Avvv",
55         "product_name": "watch"
56       },
57       "recipient": {
58         "recipient_id": "bcd85a89090745269c2e85503c20ac50",
59         "recipient_name": "Vamsi"
60       },
61       "sender": {
62         "sender_id": "f4316421ae5b40768f91a5f42f6ae24d",
63         "sender_name": "Ruban"
64       },
65       "time": "25/04/2022 22:04:50"
66     }
67   ],
68   "length": 3
69 }
70 ]
71 }
72 }
73 }
74 }
75 }
76 }
```

Console

Incorrect Product Code

Cryptography (2021-22, Sem-2)

NewImport

GET Get users

GET Get products

POST Create Transac

GET Mine Block

GET Get Blockchain

+

...

No Environment

blockchain-project / blockchain-implementation / Create Transaction

Save

Send

POST

http://localhost:5000/transactions/new

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

4

5

6

sender_id": "f4316421ae5b40768f91a5f42f6ae24d",

recipient_id": "bcd85a89890745269c2e85503c20ac50",

product_code": "eWTA0G5v5w1i13xUbmh",

price": 6000

Incorrect Product Code Entered

Body

Cookies

Headers (4)

Test Results

Status: 201 CREATED

Time: 8 ms

Size: 204 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

message": "Transaction will be added to Block None"

Console

Cookies

Bootcamp

Auto-select agent

Runner

Trash

List of Suspicious Users

Cryptography (2021-22, Sem-2)

NewImport

GET Get users

GET Get products

POST Create Tra

GET Mine Block

GET Get Blockchain

GET Get suspicious

+

...

No Environment

blockchain-project / blockchain-Implementation / Get suspicious users

Save

Send

GET

http://localhost:5000/suspicious_users

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

KEY

VALUE

DESCRIPTION

...

Bulk Edit

Key

Value

Description

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 6 ms

Size: 225 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

suspicious_users": [

{

"ID": "f4316421ae5b40768f91a5f42f6ae24d",

"Name": "Ruban"

}

]

If the transaction has an incorrect product code, the seller is added to the list of suspicious users for selling a counterfeit product

Console

Cookies

Bootcamp

Auto-select agent

Runner

Trash