# 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 $x$.

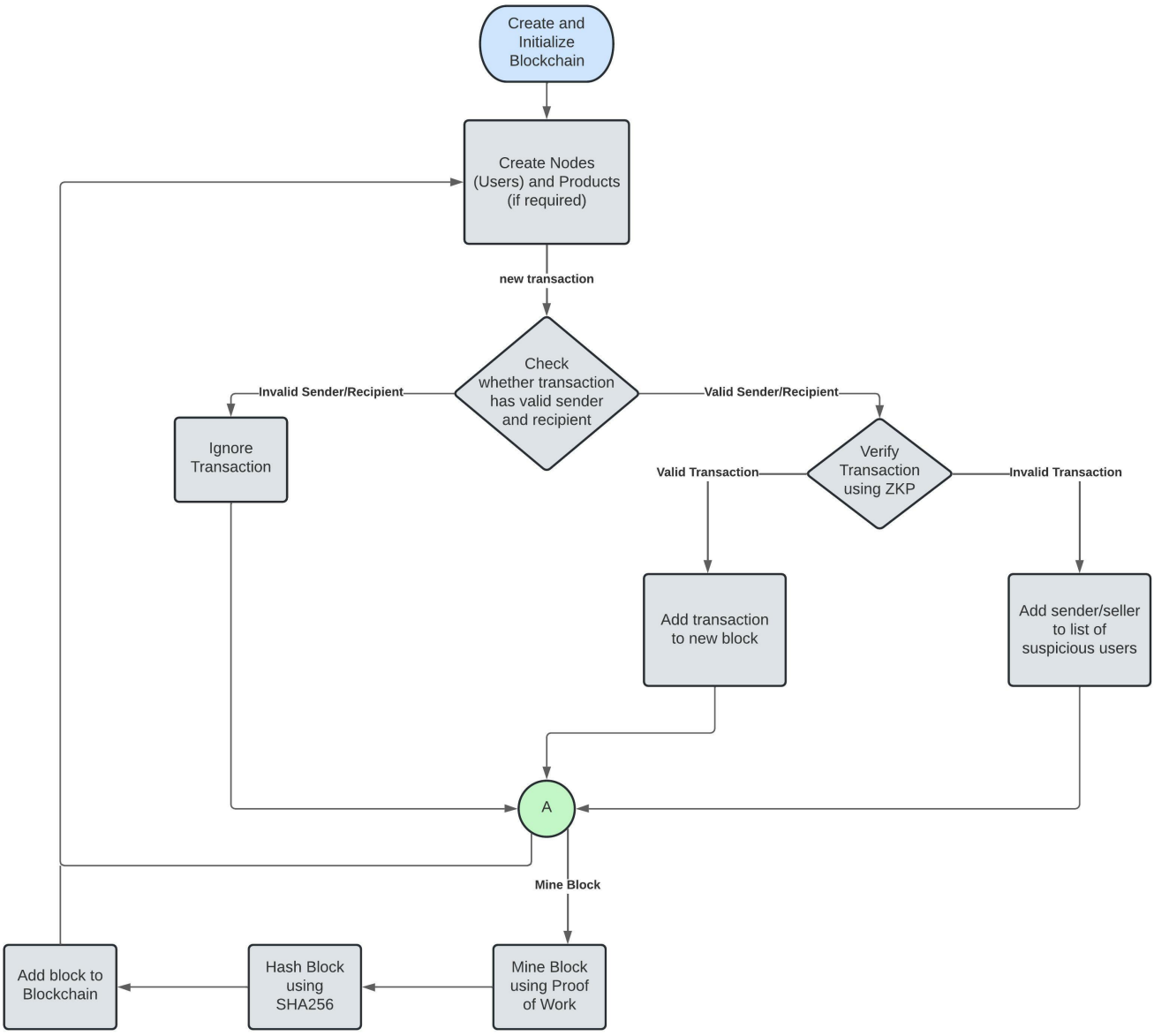
1. Alice chooses a random number $r < p - 1$ and sends it to Bob as $h = g^r \mod p$
2. Bob receives $h$ and sends back a random bit $b$ $(0/1)$.
3. Alice sends $s = (r + bx) \mod (p - 1)$ to Bob.
4. Bob computes $g^s \mod p$ which should equal $hy^b \mod p$

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

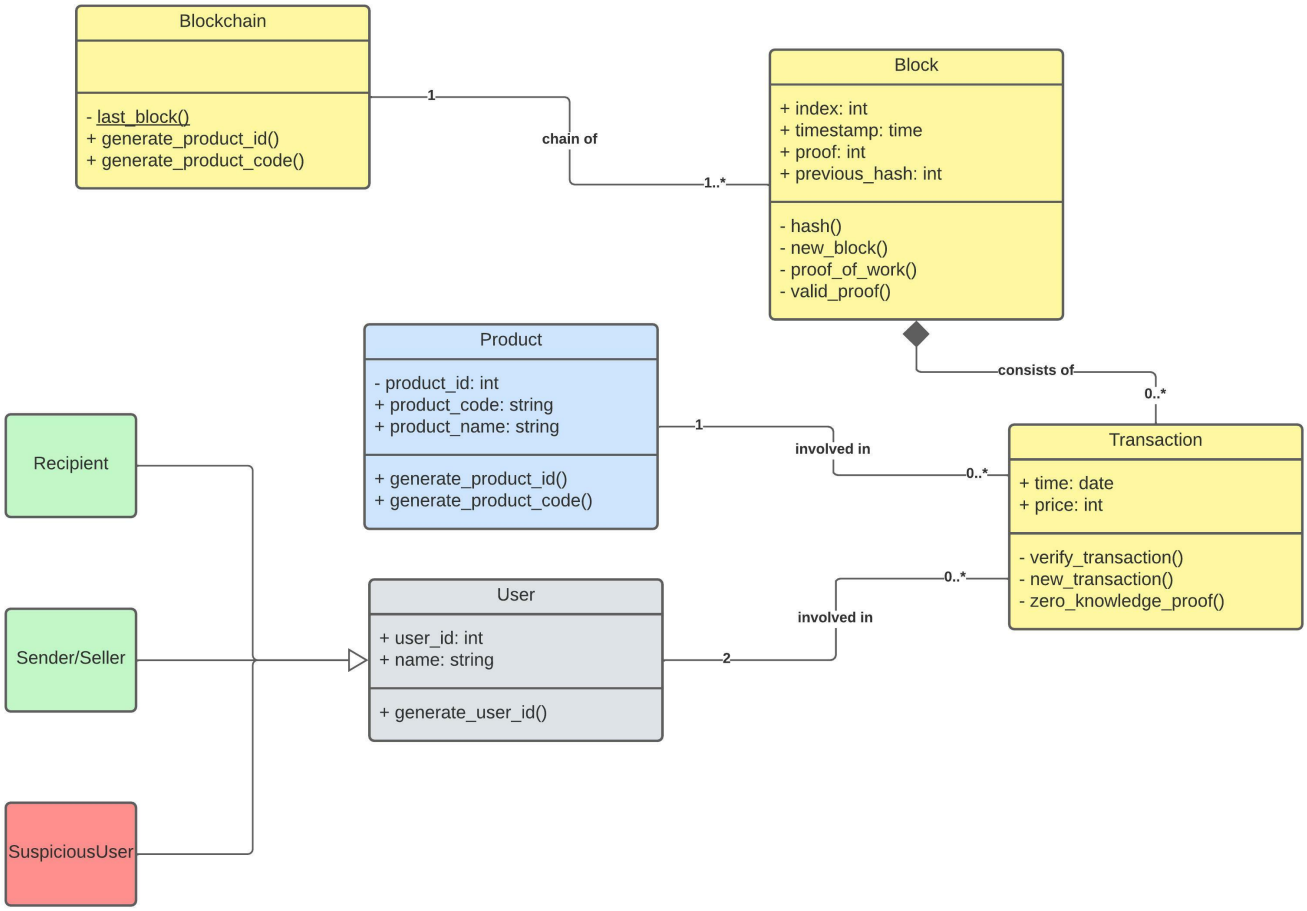
## Implementation of Zero-Knowledge Proof

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

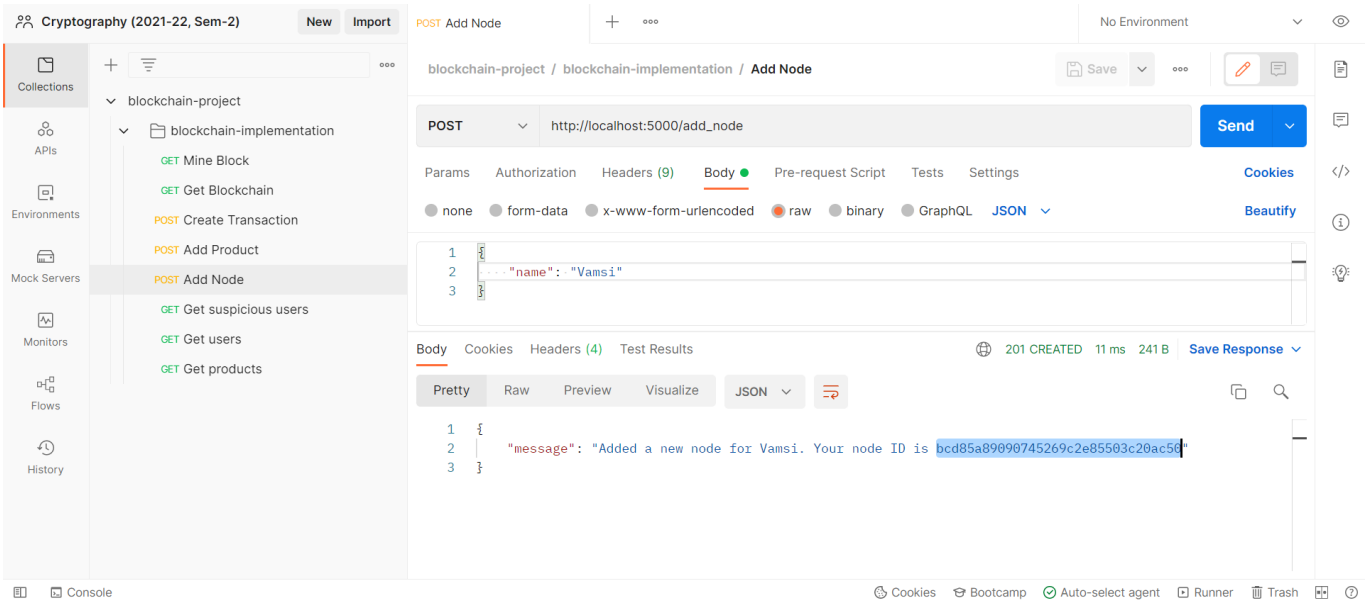


# UML Diagram of classes used

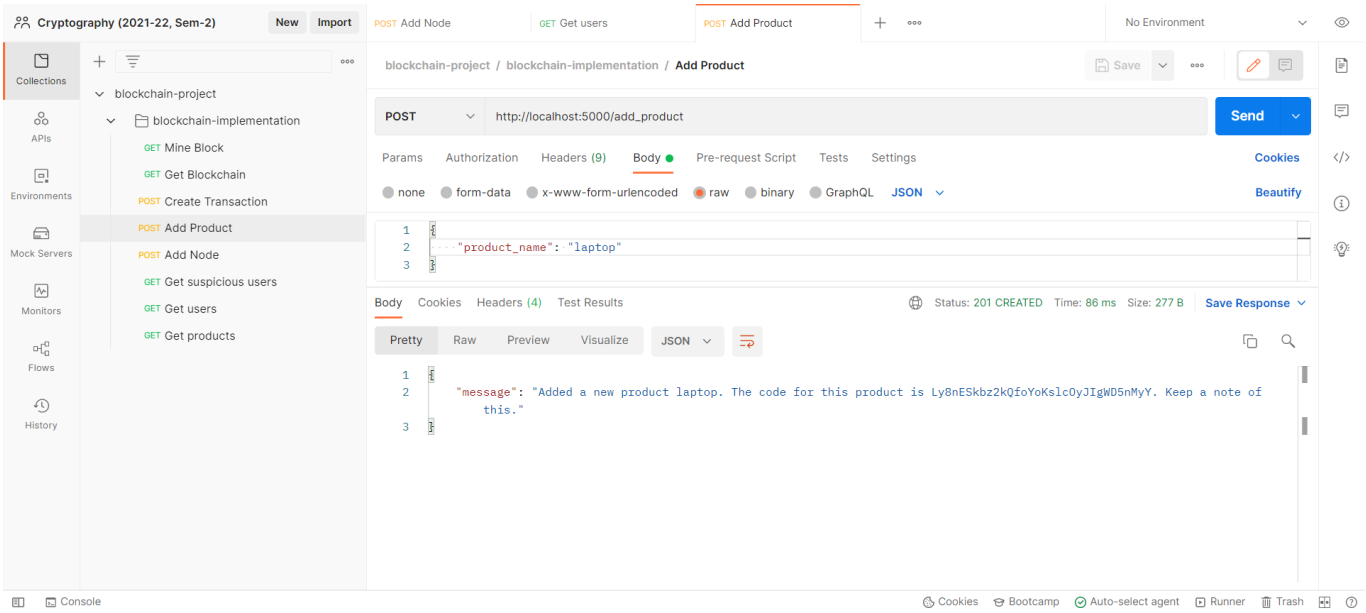



# Screenshots of the working Application

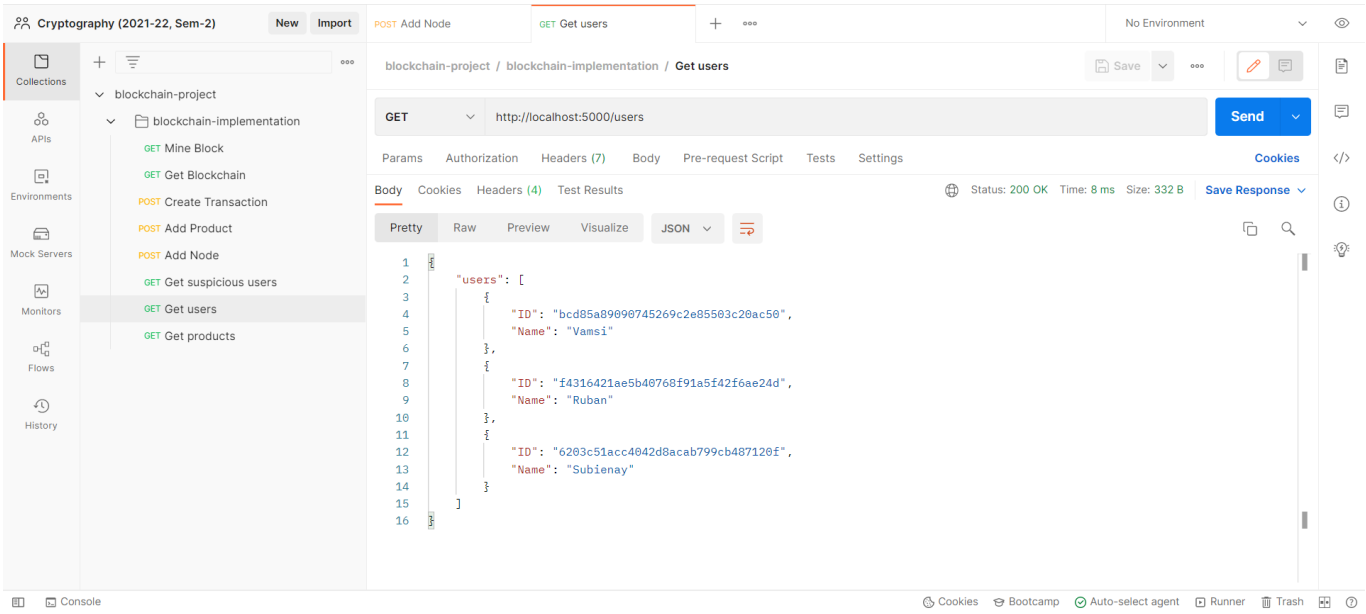
## Add Node (User) to the Blockchain



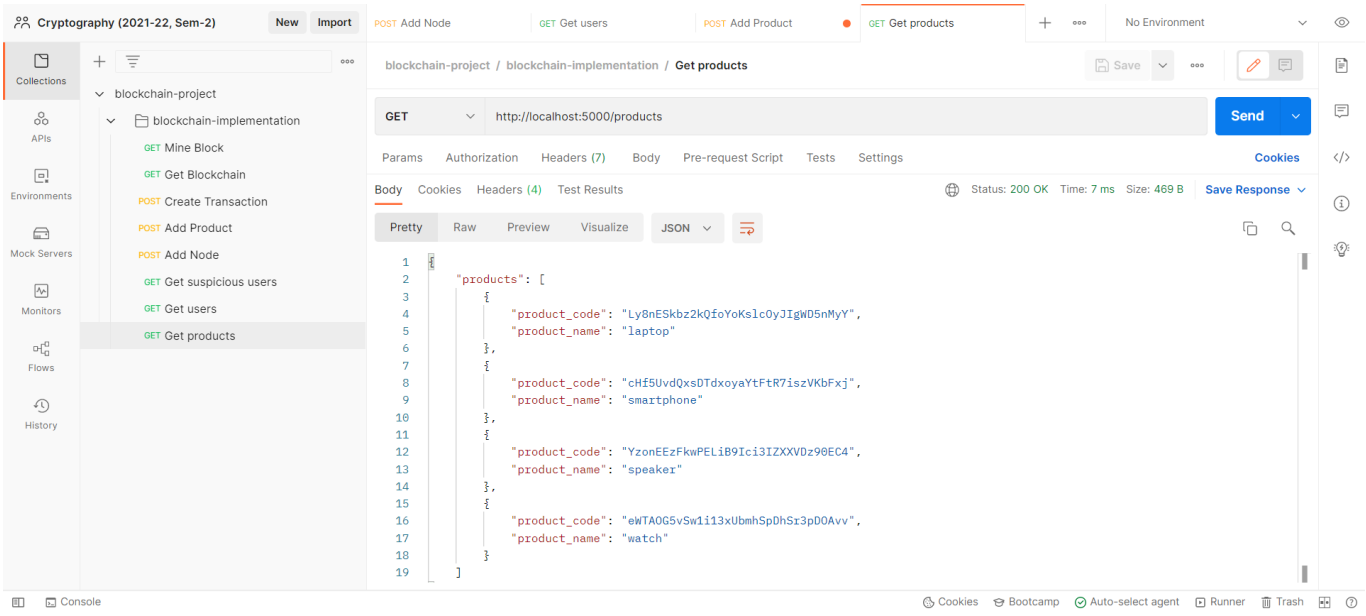
## Add a Product to a Blockchain



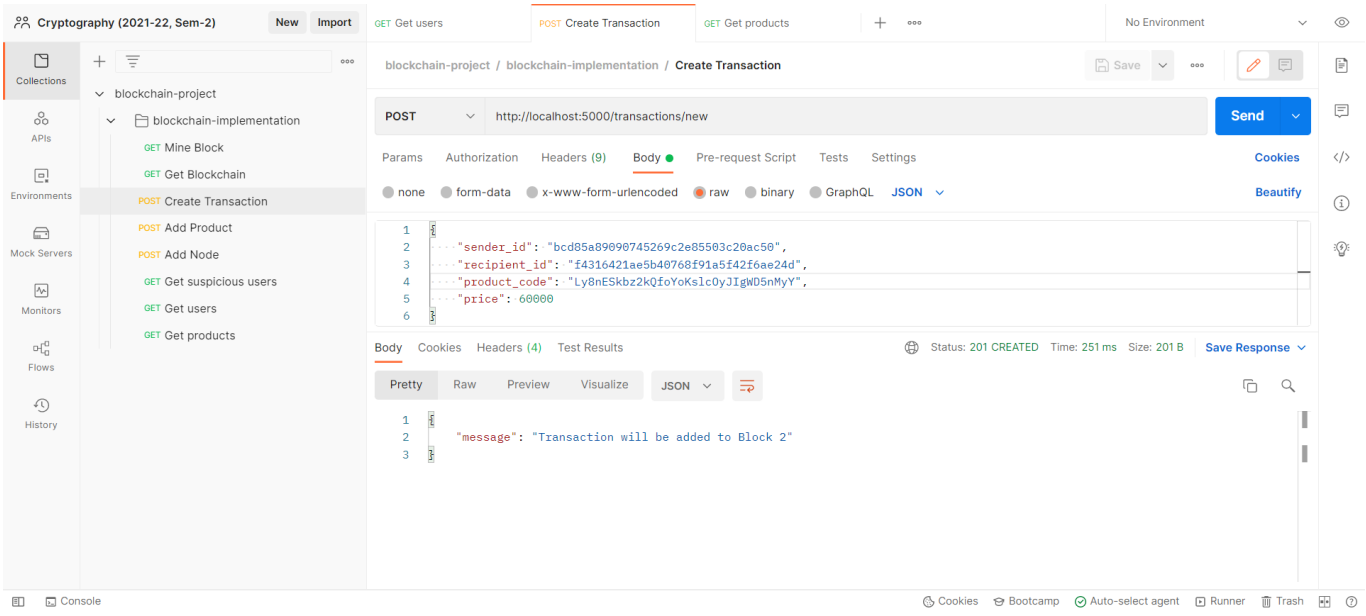

## Show all users of the Blockchain



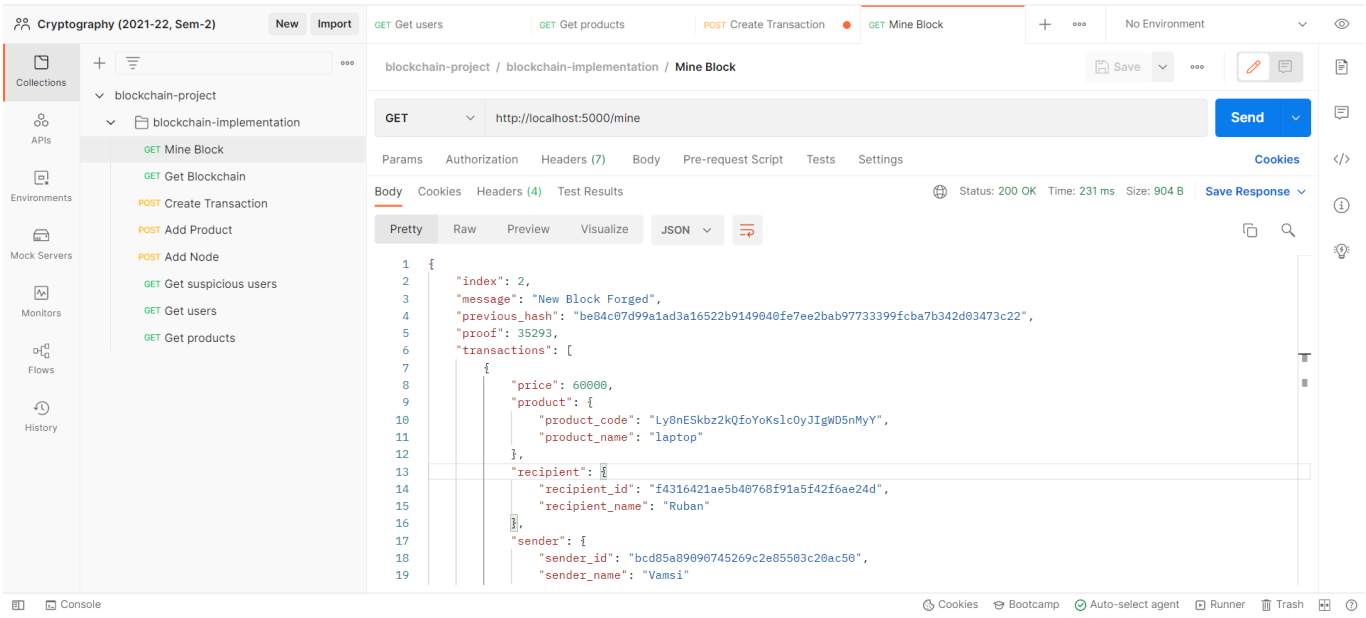
## Show all products in the Blockchain




## Create a Transaction

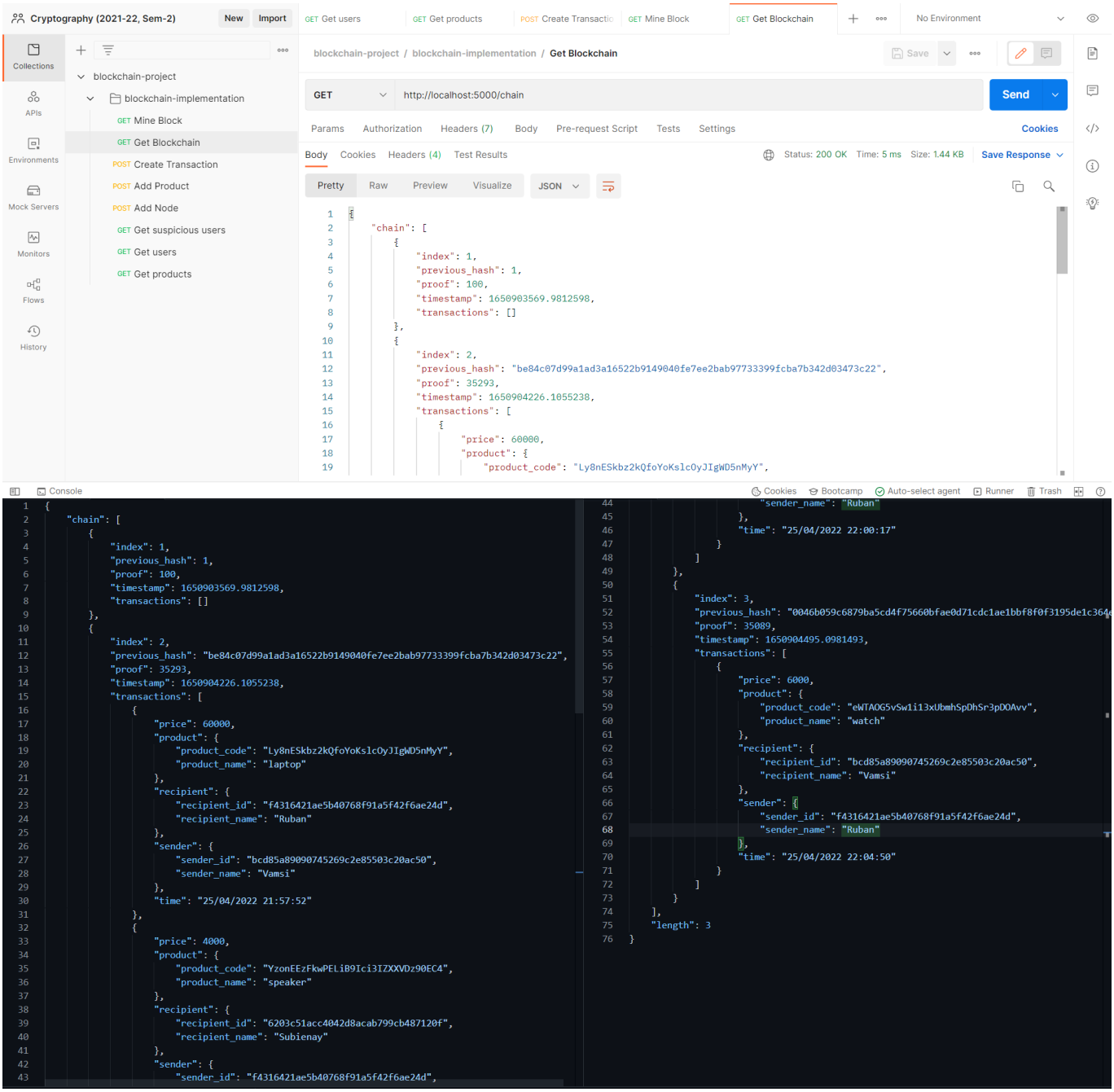


# Mine a block

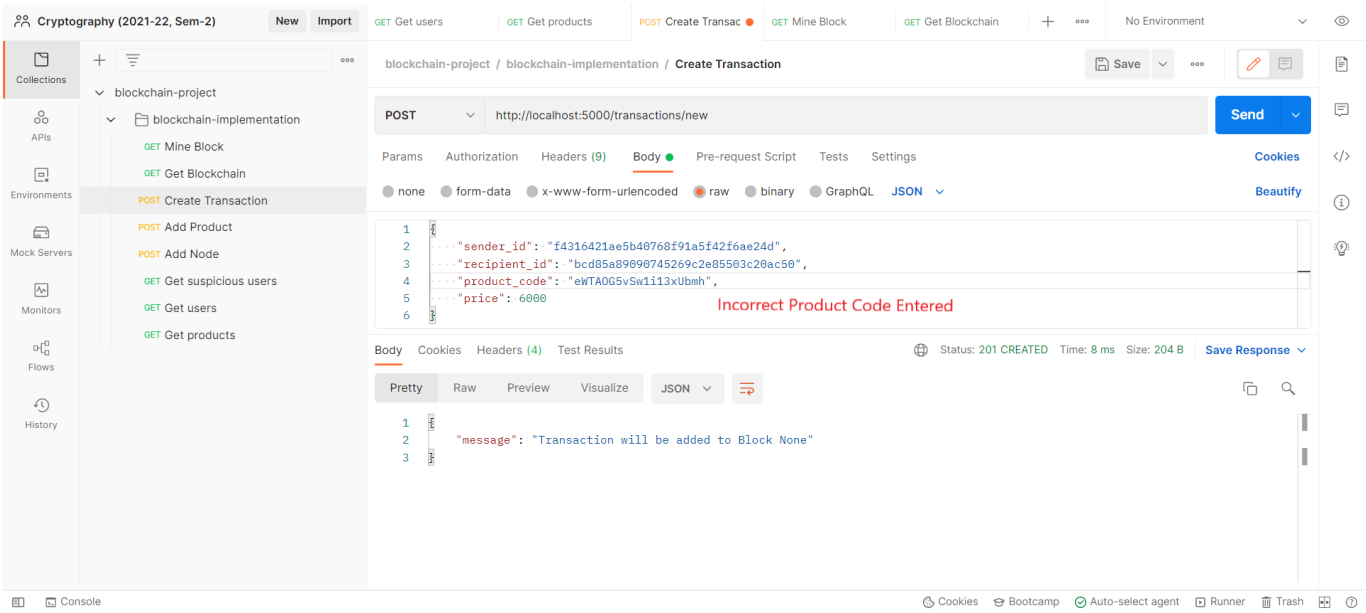

Cryptography (2021-22, Sem-2)
New
Import
GET Get users
GET Get products
POST Create Transaction
GET Mine Block
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
blockchain-project
blockchain-implementation
GET Mine Block
GET Get Blockchain
POST Create Transaction
POST Add Product
POST Add Node
GET Get suspicious users
GET Get users
GET Get products
blockchain-project / blockchain-implementation / Mine Block
Save
GET
http://localhost:5000/mine
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Body
Cookies
Headers (4)
Test Results
Status: 200 OK
Time: 231 ms
Size: 904 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"index": 2,
"message": "New Block Forged",
"previous_hash": "be84c07d99a1ad3a16522b9149040fe7ee2bab97733399fcba7b342d03473c22",
"proof": 35293,
"transactions": [
{
"price": 60000,
"product": {
"product_code": "Ly8nESkbz2kQfoYoKslcOyJIgWD5nMyY",
"product_name": "laptop"
},
"recipient": {
"recipient_id": "f4316421ae5b40768f91a5f42f6ae24d",
"recipient_name": "Ruban"
},
"sender": {
"sender_id": "bcd85a89090745269c2e85503c20ac50",
"sender_name": "Vamsi"
Console
Cookies
Bootcamp
Auto-select agent
Runner
Trash

```
{
    "index": 2,
    "message": "New Block Forged",
    "previous_hash": "be84c07d99a1ad3a16522b9149040fe7ee2bab97733399fcba7b342d03473c22",
    "proof": 35293,
    "transactions": [
        {
            "price": 60000,
            "product": {
                "product_code": "Ly8nESkbz2kQfoYoKslcOyJIgWD5nMyY",
                "product_name": "laptop"
            },
            "recipient": {
                "recipient_id": "f4316421ae5b40768f91a5f42f6ae24d",
                "recipient_name": "Ruban"
            },
            "sender": {
                "sender_id": "bcd85a89090745269c2e85503c20ac50",
                "sender_name": "Vamsi"
            },
            "time": "25/04/2022 21:57:52"
        },
        {
            "price": 4000,
            "product": {
                "product_code": "YzonEEzFkwPELiB9Ici3IZXXVDz90EC4",
                "product_name": "speaker"
            },
            "recipient": {
                "recipient_id": "6203c51acc4042d8acab799cb487120f",
                "recipient_name": "Subienay"
            },
            "sender": {
                "sender_id": "f4316421ae5b40768f91a5f42f6ae24d",
                "sender_name": "Ruban"
            },
            "time": "25/04/2022 22:00:17"
        }
    ]
}
```

# Show Blockchain



## Incorrect Product Code




## List of Suspicious Users