**Program:**

import hashlib

import json

from time import time

class Blockchain(object):

    def \_\_init\_\_(self):

        self.chain = []

        self.pending\_transactions = []

        self.new\_block(proof=88)

# Create a new block listing key/value pairs of block information in a JSON object. Reset the list of pending transactions & append the newest block to the chain.

    def new\_block(self, proof, previous\_hash=None):

        block = {

            'index': len(self.chain) + 1,

            'proof': proof,

        }

        self.chain.append(block)

        return block

#Search the blockchain for the most recent block.

    @property

    def last\_block(self):

        return self.chain[-1]

# Add a transaction with relevant info to the 'blockpool' - list of pending tx's.

    def new\_transaction(self, sender, recipient):

        transaction = {

            'sender': sender,

            'recipient': recipient,

        }

        return self.last\_block['index'] + 1

# receive one block. Turn it into a string, turn that into Unicode (for hashing). Hash with SHA256 encryption, then translate the Unicode into a hexidecimal string.

    def hash(self, block):

        string\_object = json.dumps(block, sort\_keys=True)

        block\_string = string\_object.encode()

        raw\_hash = hashlib.sha256(block\_string)

        hex\_hash = raw\_hash.hexdigest()

        return hex\_hash

blockchain = Blockchain()

t1 = blockchain.new\_transaction("Alaikya", "Santosh")

t2 = blockchain.new\_transaction("Santosh", "Sudhakar")

t3 = blockchain.new\_transaction("Sudhakar", "Walchand")

blockchain.new\_block(789)

t4 = blockchain.new\_transaction("Walchand", "Alaikya")

t5 = blockchain.new\_transaction("Alaikya", "Savita")

t6 = blockchain.new\_transaction("Savita", "Santosh")

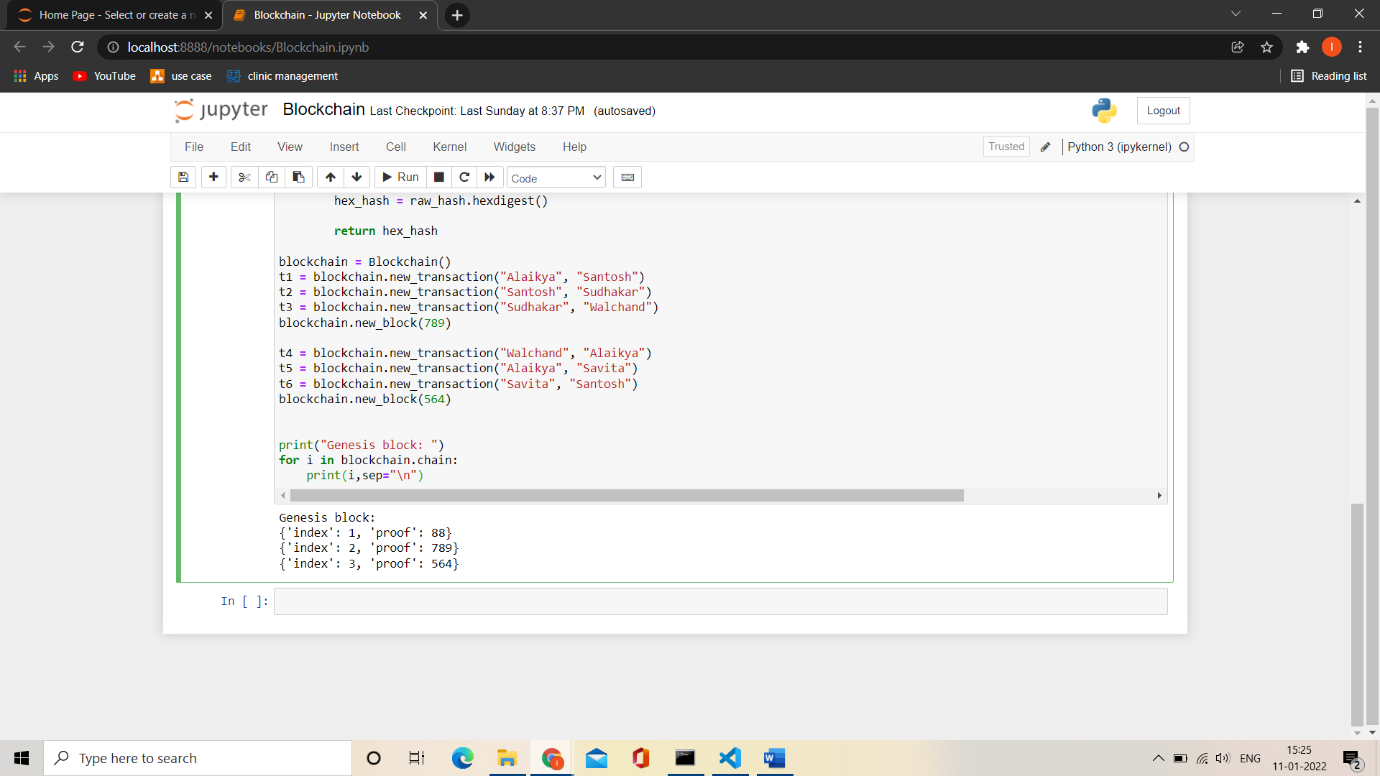
blockchain.new\_block(564)

print("Genesis block: ")

for i in blockchain.chain:

    print(i,sep="\n")

**Screenshots:**

****