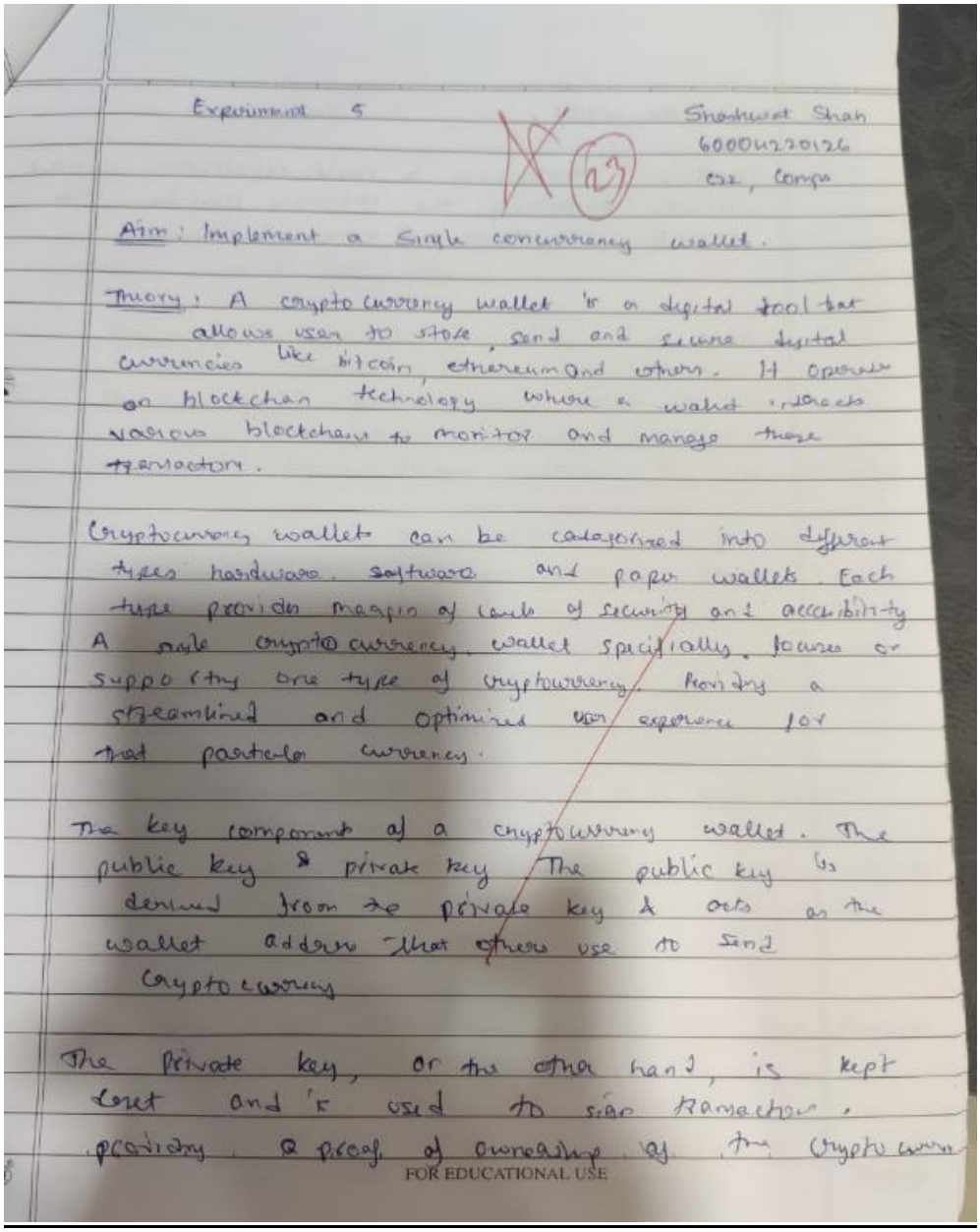
BLOCKCHAIN TECHNOLOGY

EXPERIMENT NO.05



Experiment 5

Sneshwar Shah
60004220126
CE2, Comps

Aim: Implement a Single concurrency wallet.

Theory: A crypto currency wallet is a digital tool that allows user to store, send and secure digital currencies like bitcoin, ethereum and others. It operates on blockchain technology where a wallet interacts various blockchain to monitor and manage those transactions.

Cryptocurrency wallets can be categorized into different types hardware, software and paper wallets. Each type provides margin of level of security and accessibility. A single crypto currency wallet specifically, focuses or supports one type of cryptocurrency. Providing a streamlined and optimized user experience for that particular currency.

The key components of a cryptocurrency wallet. The public key & private key. The public key is derived from the private key & acts as the wallet address that others use to send cryptocurrency.

The private key, on the other hand, is kept secret and is used to sign transactions, providing a proof of ownership of the crypto curr...

CODE & OUTPUT :-

```
import secrets
from eth_account import Account
from web3 import Web3

class SimpleEthereumWallet:
    def __init__(self):
        self.w3 = Web3(Web3.HTTPProvider('https://mainnet.infura.io/v3/****'))
        self.account = None

    def create_wallet(self):
        private_key = '0x' + secrets.token_hex(32)
        self.account = Account.from_key(private_key)
        return {
            'address': self.account.address,
            'private_key': private_key
        }

    def import_wallet(self, private_key):
        self.account = Account.from_key(private_key)
        return self.account.address

    def get_balance(self):
        if not self.account:
            return "Wallet not initialized"
        balance_wei = self.w3.eth.get_balance(self.account.address)
        balance_eth = self.w3.from_wei(balance_wei, 'ether')
        return f"{balance_eth} ETH"

    def send_transaction(self, to_address, amount_eth):
        if not self.account:
            return "Wallet not initialized"

        nonce = self.w3.eth.get_transaction_count(self.account.address)
        gas_price = self.w3.eth.gas_price

        tx = {
            'nonce': nonce,
            'to': to_address,
            'value': self.w3.to_wei(amount_eth, 'ether'),
            'gas': 21000,
            'gasPrice': gas_price
        }

        signed_tx = self.account.sign_transaction(tx)
        tx_hash = self.w3.eth.send_raw_transaction(signed_tx.rawTransaction)
```

```
        return self.w3.to_hex(tx_hash)

# Usage example
wallet = SimpleEthereumWallet()

# Create a new wallet
new_wallet = wallet.create_wallet()
print("New wallet created:")
print(f"Address: {new_wallet['address']}")
print(f"Private Key: {new_wallet['private_key']}")

# Import an existing wallet
imported_address = wallet.import_wallet(new_wallet['private_key'])
print(f"\nImported wallet address: {imported_address}")

# Get balance
balance = wallet.get_balance()
print(f"Wallet balance: {balance}")
```