

BLOCKCHAIN TECHNOLOGY

EXPERIMENT NO.05

Experiment 5

Shashwat Shah

60001220126

CSE, Compu

Aim: Implement a Single currency wallet.

Theory: A cryptocurrency wallet is a digital tool that allows users to store, send and receive digital currencies like bitcoin, ethereum and others. It operates on blockchain technology where a wallet interacts with various blockchains to monitor and manage these transactions.

Cryptocurrency wallets can be categorized into different types: hardware, software and paper wallets. Each type provides varying levels of security and accessibility. A multi-currency wallet specifically focuses on supporting one type of cryptocurrency, providing a streamlined and optimized user experience for that particular currency.

The key components of a cryptocurrency wallet are 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 cryptocurrency.

FOR EDUCATIONAL USE

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}")

```

The screenshot shows a VS Code editor with the file `CryptoWallet.py` open. The code in the file is as follows:

```

12 class SimpleEthereumWallet:
13     def create_wallet(self):
21         'address': self.account.address,
22         'private_key': private_key
23     }
24
25     def import_wallet(self, private_key):
26         self.account = Account.from_key(private_key)
27         return self.account.address
28
29     def get_balance(self):
30         if not self.account:
31             return "Wallet not initialized"
32         balance_wei = self.w3.eth.get_balance(self.account.address)
33         balance_eth = self.w3.from_wei(balance_wei, 'ether')
34         return f"(balance_eth) ETH"
35

```

The terminal output shows the following steps:

```

Downloaded frozenlist-1.4.1-cp310-cp310-win_amd64.whl (50 kB)
Downloaded multidict-6.0.5-cp310-cp310-win_amd64.whl (28 kB)
Downloaded yarl-1.9.7-cp310-cp310-win_amd64.whl (108 kB)
Building wheels for collected packages: pyunormalize
  Building wheel for pyunormalize (pyproject.toml) ... done
  Created wheel for pyunormalize: filename=pyunormalize-15.1.0-py3-none-any.whl size=516052 sha256=f7d941d8bb7faa88e0daf08ae3c1eb921c87d8ae7b74ba94e19e2bca215c2
  Stored in directory: c:\users\djsce.student\appdata\local\pip\cache\wheels\b8\b6\c6\6fc11f58d3fd02f387cf3b18479bb5cd4143995d0fa401622
Successfully built pyunormalize
Installing collected packages: websockets, types-requests, pyunormalize, multidict, frozenlist, attrs, async-timeout, aiohappyeyeballs, yarl, aiohttp, aiohttp, web3
Successfully installed aiohappyeyeballs-2.4.0 aiohttp-3.10.5 aiosignal-1.3.1 async-timeout-4.0.3 attrs-24.2.0 frozenlist-1.4.1 multidict-6.0.5 pyunormalize-15.1.0 types-requests-2.32.0.20240712 web3-7.2.0 websockets-13.0.1 yarl-1.9.7
PS C:\QNS & C:\Users\djsce.student\AppData\Local\Microsoft\WindowsApps\python3.10.exe c:\Users\djsce.student\Desktop\CryptoWallet.py
New wallet created:
Address: 0xbda2356832a7c6d46fc56904a6e0b9386126dc76b
Private Key: 0x9bda28872d6736a68d26f3c07cca7865a30da89c7cc8e196a9f5a4dad7511e
Imported wallet address: 0xbda2356832a7c6d46fc56904a6e0b9386126dc76b
Wallet balance: 0 ETH

```