Department of Computer Engineering

Academic Term: Jan-May 23-24

Class: B.E Computer Sem -VII

Subject: Blockchain Technology Lab

Subject Code : CSDL7022

Practical No:	**5**
Title:	**Embedding wallet and transaction using Solidity**
Date of Performance:	**25/08/2023**
Date of Submission:	**25/08/2023**
Roll No:	**9427**
Name of the Student:	**Atharva Prashant Pawar**

Evaluation:

Sr. No	**Rubric**	**Grade**
1	**Time Line (2)**	
2	**Output (3)**	
3	**Code optimization (2)**	
4	**Post lab (3)**	

Signature of the Teacher :

Experiment No. 5

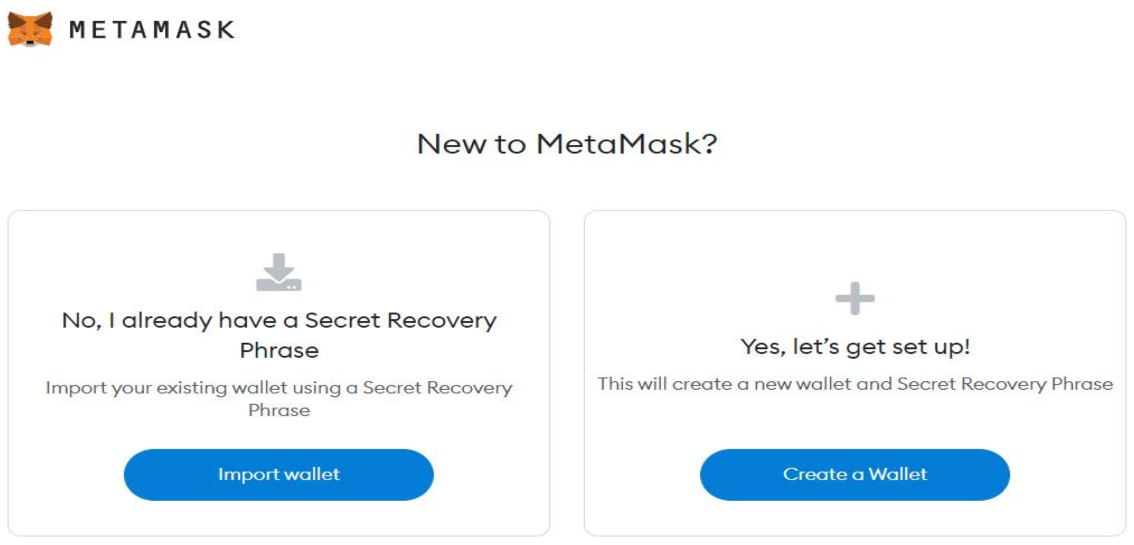
Embedding wallet and transaction using Solidity

Aim: Creating a smart contract using Metamask and performing the transactions.

Theory:

Step 1: Create a wallet at meta-mask

- Install MetaMask in Chrome browser and enable it.


METAMASK

New to MetaMask?

No, I already have a Secret Recovery Phrase

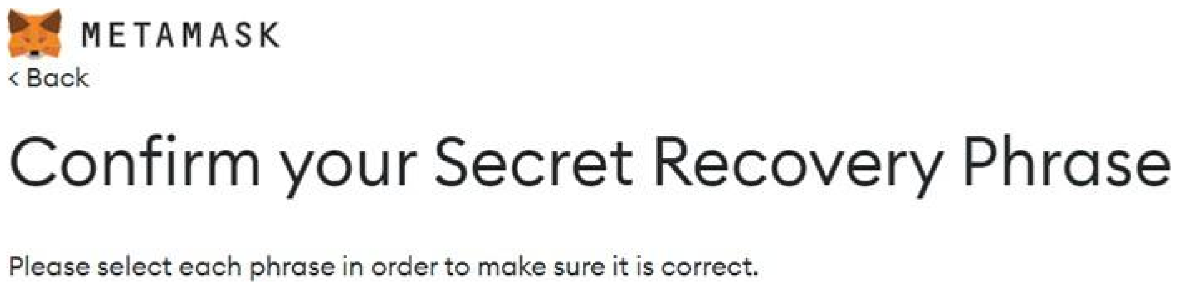
Import your existing wallet using a Secret Recovery Phrase

Import wallet

Yes, let's get set up!

This will create a new wallet and Secret Recovery Phrase

Create a Wallet


METAMASK

< Back

Confirm your Secret Recovery Phrase

Please select each phrase in order to make sure it is correct.

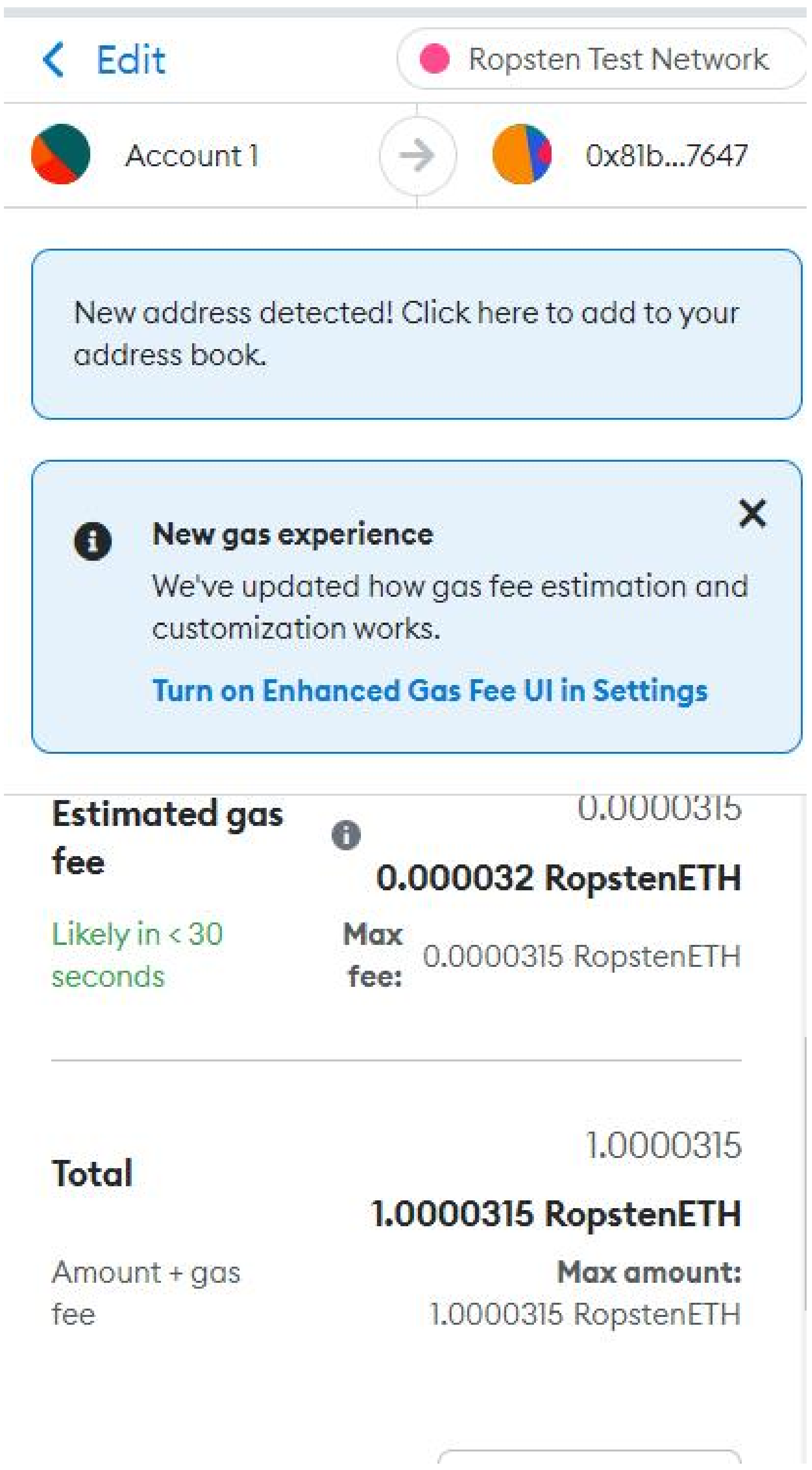
Step 2: Select any one test network

Click on show networks. Enable show networks and other information. Following test networks can be seen in your MetaMask wallet. These networks are only for the purpose of testing and ethers involved in it have no value.

- Ropsten Test Network
- Kovan Test Network

- Rinkeby Test Network
- Goerli Test Network

Select Ropsten Test Network.


< Edit
Ropsten Test Network
Account 1 → 0x81b...7647

New address detected! Click here to add to your address book.

New gas experience
We've updated how gas fee estimation and customization works.
Turn on Enhanced Gas Fee UI in Settings

Estimated gas fee 0.0000315
0.000032 RopstenETH
Likely in < 30 seconds
Max fee: 0.0000315 RopstenETH

Total 1.0000315
1.0000315 RopstenETH
Amount + gas fee
Max amount: 1.0000315 RopstenETH

Step 3: Add some dummy Ethers to your wallet

- To test the smart contract, your MetaMask wallet should contain some dummy ethers. For example, if you test a contract using the Ropsten test network, select it and you will find 0 ETH as the initial balance in your account. Click on Buy button. Buy test ethers.

Step 4: Compile and Deploy following contract in Ethereum Remix IDE. Give name to file token.sol. Deploy by selecting **QKCToken–contract/token.sol**. **Select Environment as Injected Provider-Metamask.** Replace **YOUR_METAMASK_WALLET_ADDRESS** in code by your wallet address which you have created.

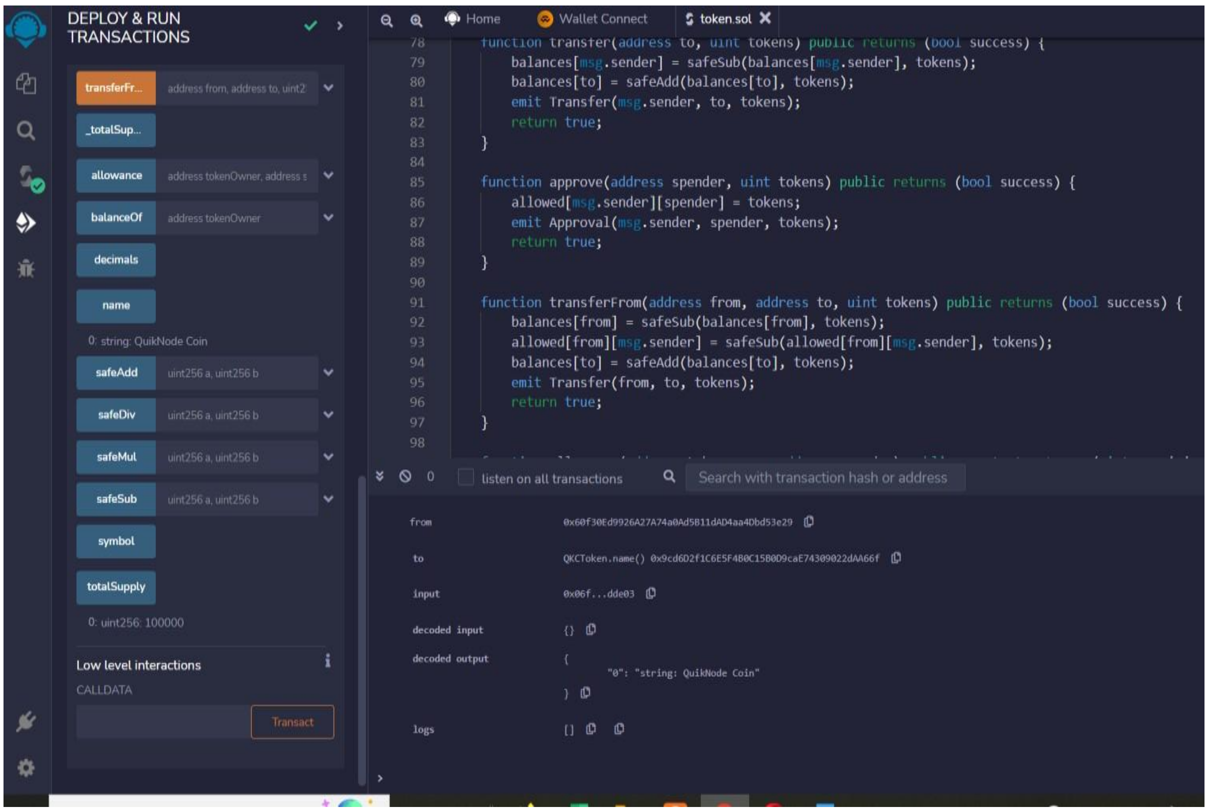
```
pragma solidity ^0.8.08;
//Safe Math Interface
contract SafeMath {
   function safeAdd(uint a, uint b) public pure returns (uint c)
      {c = a + b;
      require(c >= a);
   }

   function safeSub(uint a, uint b) public pure returns (uint c)
      {require(b <= a);
      c = a - b;
   }
   function safeMul(uint a, uint b) public pure returns (uint c)
      {c = a * b;
      require(a == 0 || c / a == b);
   }
   function safeDiv(uint a, uint b) public pure returns (uint c)
      {require(b > 0);
      c = a / b;
   }
}

//ERC Token Standard #20 Interface
contract ERC20Interface {
   function totalSupply() public constant returns (uint);
   function balanceOf(address tokenOwner) public constant returns (uint balance);
   function allowance(address tokenOwner, address spender) public constant returns (uint
remaining);
   function transfer(address to, uint tokens) public returns (bool success);
   function approve(address spender, uint tokens) public returns (bool success);
   function transferFrom(address from, address to, uint tokens) public returns (bool success);

   event Transfer(address indexed from, address indexed to, uint tokens);
   event Approval(address indexed tokenOwner, address indexed spender, uint tokens);
}

//Contract function to receive approval and execute function in one call
contract ApproveAndCallFallBack {
   function receiveApproval(address from, uint256 tokens, address token, bytes data) public;
}
```

```
//Actual token contract

contract QKCToken is ERC20Interface, SafeMath
   {string public symbol;
   string public  name;
   uint8 public decimals;
   uint public _totalSupply;

   mapping(address => uint) balances;
   mapping(address => mapping(address => uint)) allowed;

   constructor() public
      {symbol = "QKC";
      name = "QuikNode Coin";
      decimals = 2;
      _totalSupply = 100000;
      balances[YOUR_METAMASK_WALLET_ADDRESS] = _totalSupply;
      emit Transfer(address(0), YOUR_METAMASK_WALLET_ADDRESS, _totalSupply);
   }

   function totalSupply() public constant returns (uint)
      {return _totalSupply - balances[address(0)];
   }

   function balanceOf(address tokenOwner) public constant returns (uint balance)
      {return balances[tokenOwner];
   }

   function transfer(address to, uint tokens) public returns (bool success)
      {balances[msg.sender] = safeSub(balances[msg.sender], tokens);
      balances[to] = safeAdd(balances[to], tokens);
      emit Transfer(msg.sender, to, tokens);
      return true;
   }

   function approve(address spender, uint tokens) public returns (bool success)
      {allowed[msg.sender][spender] = tokens;
      emit Approval(msg.sender, spender, tokens);
      return true;
   }

   function transferFrom(address from, address to, uint tokens) public returns (bool success)
      {balances[from] = safeSub(balances[from], tokens);
      allowed[from][msg.sender] = safeSub(allowed[from][msg.sender], tokens);
      balances[to] = safeAdd(balances[to], tokens);
      emit Transfer(from, to, tokens);
```

```
        return true;
    }

    function allowance(address tokenOwner, address spender) public constant returns (uint
remaining) {
        return allowed[tokenOwner][spender];
    }

    function approveAndCall(address spender, uint tokens, bytes data) public returns (bool
success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        ApproveAndCallFallBack(spender).receiveApproval(msg.sender, tokens, this, data);
        return true;
    }

    function () public payable
        {revert();
    }
}
```

Methods

Step 5: totalSupply: A method that defines the total supply of your tokens, When this limit is reached the smart contract will refuse to create new tokens. As defined in contract total supply is 100000 shown below.

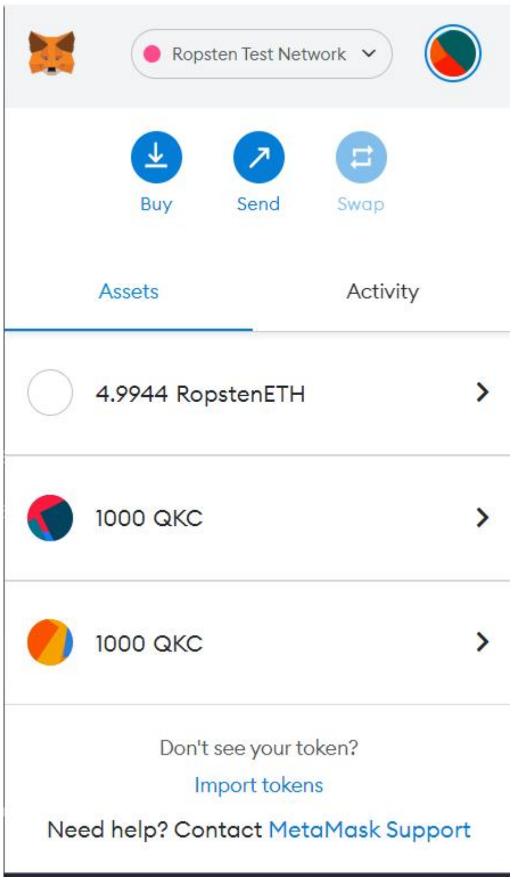


0: string: QuikNode Coin

0: uint256: 100000

```
    function transfer(address to, uint tokens) public returns (bool success) {
        balances[msg.sender] = safeSub(balances[msg.sender], tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(msg.sender, to, tokens);
        return true;
    }

    function approve(address spender, uint tokens) public returns (bool success) {
        allowed[msg.sender][spender] = tokens;
        emit Approval(msg.sender, spender, tokens);
        return true;
    }

    function transferFrom(address from, address to, uint tokens) public returns (bool success) {
        balances[from] = safeSub(balances[from], tokens);
        allowed[from][msg.sender] = safeSub(allowed[from][msg.sender], tokens);
        balances[to] = safeAdd(balances[to], tokens);
        emit Transfer(from, to, tokens);
        return true;
    }
```

```
from            0x60f30Ed9926A27A74a0Ad5B11dAD4aa4Dbd53e29
to              QKCToken.name() 0x9cd6D2f1C6E5F4B0C15B0D9caE74309022dAA66f
input           0x06f...dde03
decoded input   {}
decoded output  {
                    "0": "string: QuikNode Coin"
                }
logs            []
```


Ropsten Test Network

4.9944 RopstenETH

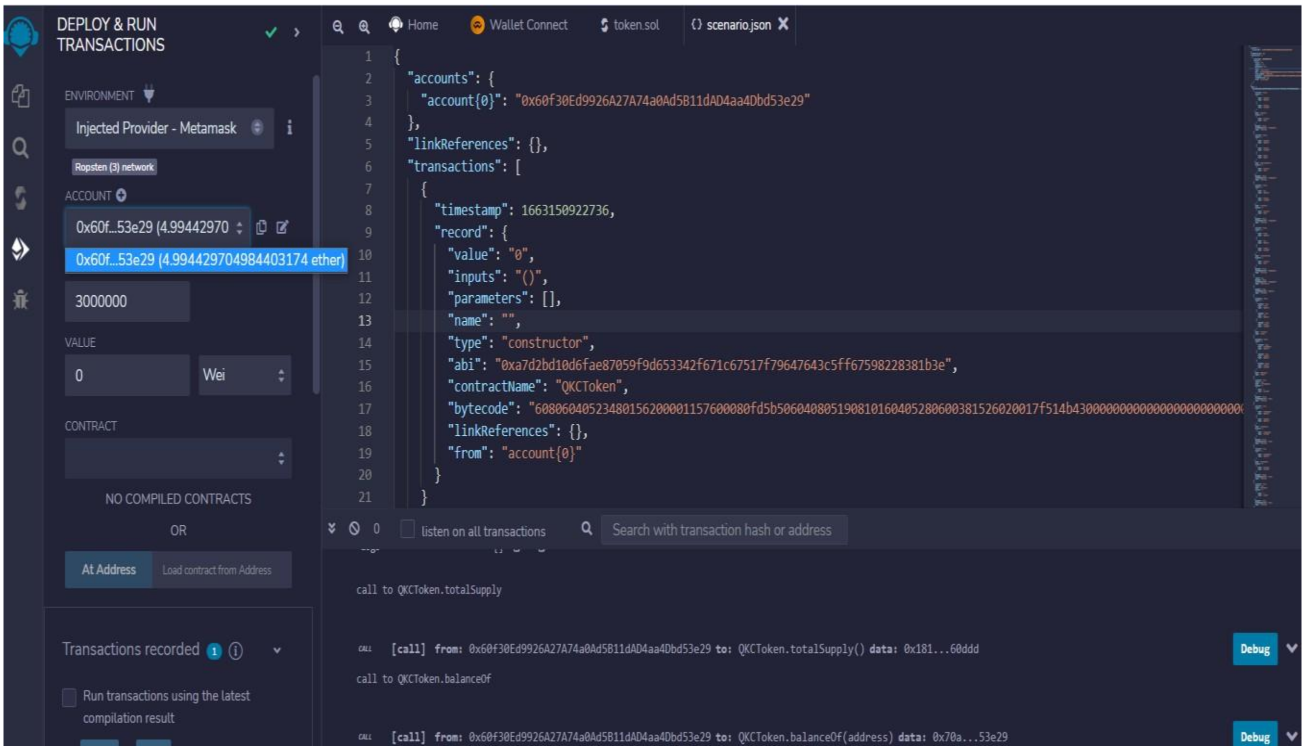
1000 QKC

1000 QKC

Don't see your token?

Import tokens

Need help? Contact MetaMask Support


DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - Metamask (Ropsten (3) network)

ACCOUNT: 0x60f...53e29 (4.99442970

0x60f...53e29 (4.994429704984403174 ether)

3000000

VALUE: 0 Wei

CONTRACT

NO COMPILED CONTRACTS

OR

At Address | Load contract from Address

Transactions recorded 1

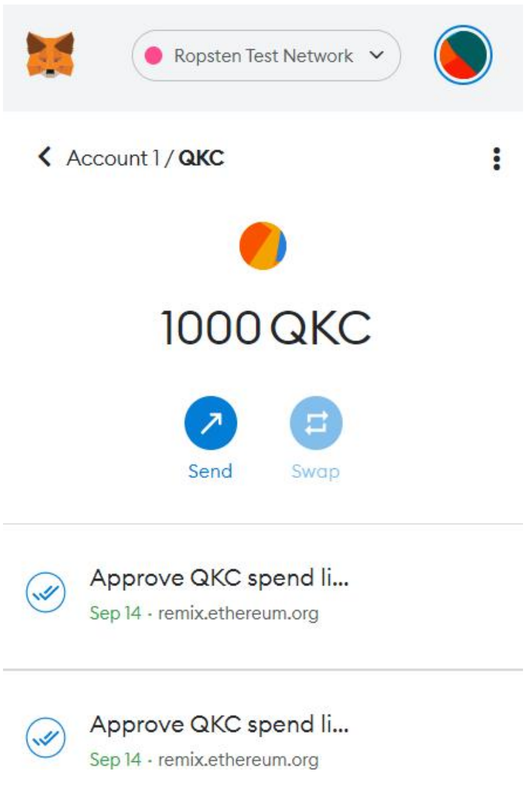
Run transactions using the latest compilation result

```
{
  "accounts": {
    "account{0}": "0x60f30Ed9926A27A74a0Ad5B11dAD4aa4Dbd53e29"
  },
  "linkReferences": {},
  "transactions": [
    {
      "timestamp": 1663150922736,
      "record": {
        "value": "0",
        "inputs": "()",
        "parameters": [],
        "name": "",
        "type": "constructor",
        "abi": "0xa7d2bd10d6fae87059f9d653342f671c67517f79647643c5ff67598228381b3e",
        "contractName": "QKCToken",
        "bytecode": "60806040523480156200011576000080fd5b50604080519081016040528060038152602001 7f514b430000000000000000000000000
        "linkReferences": {},
        "from": "account{0}"
      }
    }
```

```
call to QKCToken.totalSupply

CALL  [call] from: 0x60f30Ed9926A27A74a0Ad5B11dAD4aa4Dbd53e29 to: QKCToken.totalSupply() data: 0x181...60ddd
call to QKCToken.balanceOf

CALL  [call] from: 0x60f30Ed9926A27A74a0Ad5B11dAD4aa4Dbd53e29 to: QKCToken.balanceOf(address) data: 0x70a...53e29
```

Home
Wallet Connect
token.sol
scenario.json
listen on all transactions
Search with transaction hash or address
Debug
Debug

balanceOf: A method that returns the number of tokens a wallet address has.


Ropsten Test Network

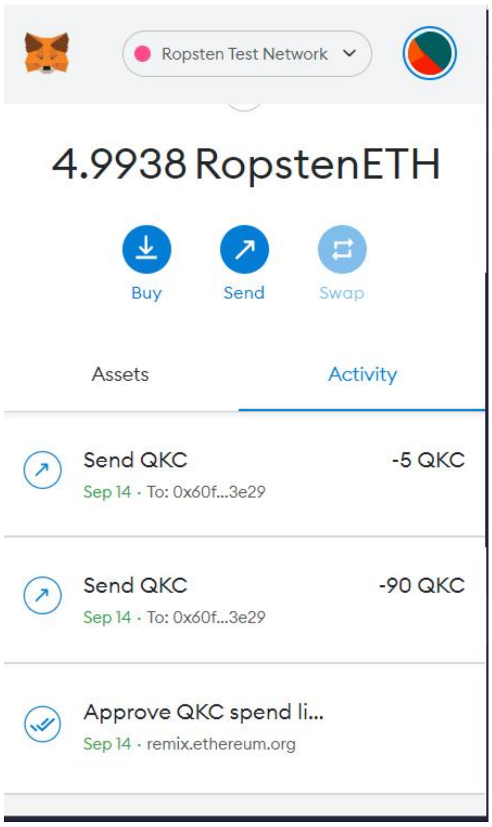
Account 1 / **QKC**

1000 QKC

Approve QKC spend li...
Sep 14 · remix.ethereum.org

Approve QKC spend li...
Sep 14 · remix.ethereum.org

transfer: A method that takes a certain amount of tokens from the total supply and gives it to a user.


Ropsten Test Network

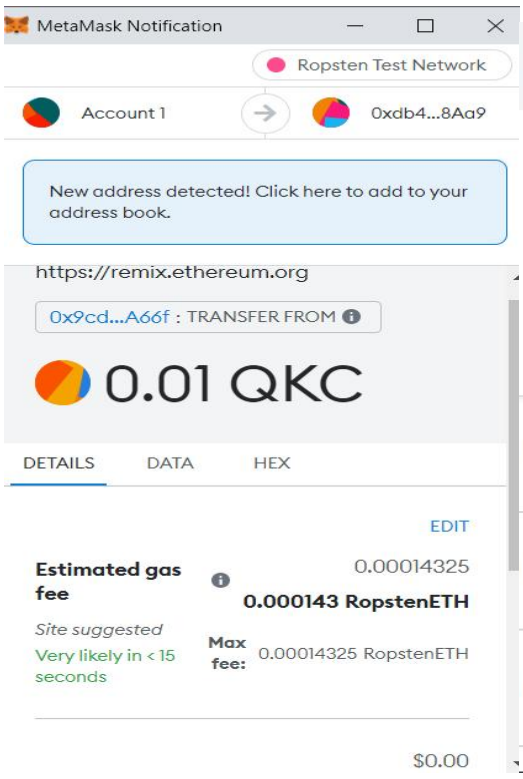
4.9938 RopstenETH

Send QKC -5 QKC
Sep 14 · To: 0x60f...3e29

Send QKC -90 QKC
Sep 14 · To: 0x60f...3e29

Approve QKC spend li...
Sep 14 · remix.ethereum.org

transferFrom: Another type of transfer method which is used to transfer tokens between users.


MetaMask Notification

Ropsten Test Network

Account 1 → 0xdb4...8Aa9

New address detected! Click here to add to your address book.

https://remix.ethereum.org

0x9cd...A66f : TRANSFER FROM

0.01 QKC

DETAILS DATA HEX

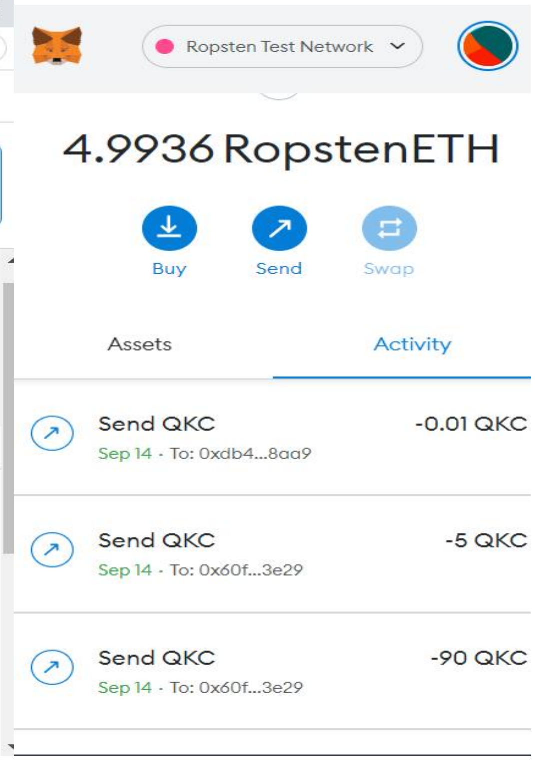
EDIT

Estimated gas fee 0.00014325
0.000143 RopstenETH

Site suggested
Very likely in < 15 seconds

Max fee: 0.00014325 RopstenETH

$0.00


Ropsten Test Network

4.9936 RopstenETH

Buy Send Swap

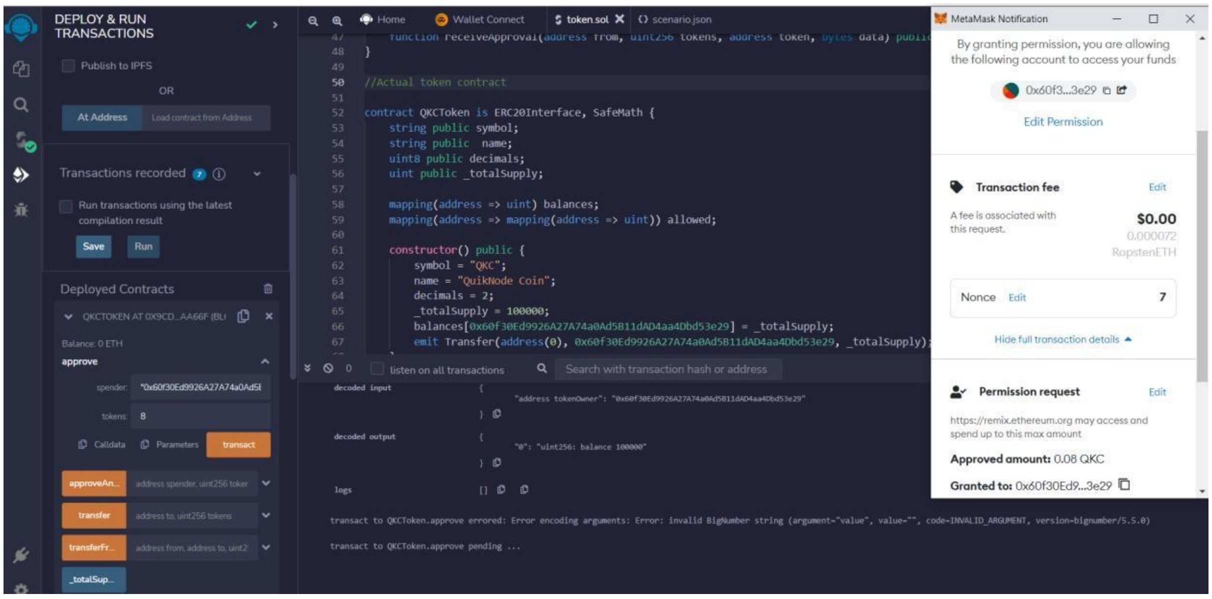
Assets Activity

Send QKC -0.01 QKC
Sep 14 · To: 0xdb4...8aa9

Send QKC -5 QKC
Sep 14 · To: 0x60f...3e29

Send QKC -90 QKC
Sep 14 · To: 0x60f...3e29

approve: This method verifies whether a smart contract is allowed to allocate a certain amount of tokens to a user, considering the total supply.


DEPLOY & RUN TRANSACTIONS

Publish to IPFS

OR

At Address | Load contract from Address

Transactions recorded 7

Run transactions using the latest compilation result

Save Run

Deployed Contracts

QKCTOKEN AT 0X9CD...AA66F (BL

Balance: 0 ETH

approve

spender: "0x60f30Ed9926A27A74a0Ad5I

tokens: 8

Calldata Parameters transact

approveAn... address spender, uint256 toker

transfer address to, uint256 tokens

transferFr... address from, address to, uint2

_totalSup...

Home | Wallet Connect | token.sol | scenario.json

```
    function receiveApproval(address from, uint256 tokens, address token, bytes data) publi
}

//Actual token contract

contract QKCToken is ERC20Interface, SafeMath {
    string public symbol;
    string public  name;
    uint8 public decimals;
    uint public _totalSupply;

    mapping(address => uint) balances;
    mapping(address => mapping(address => uint)) allowed;

    constructor() public {
        symbol = "QKC";
        name = "QuikNode Coin";
        decimals = 2;
        _totalSupply = 100000;
        balances[0x60f30Ed9926A27A74a0Ad5B11dAD4aa4Dbd53e29] = _totalSupply;
        emit Transfer(address(0), 0x60f30Ed9926A27A74a0Ad5B11dAD4aa4Dbd53e29, _totalSupply);
```

listen on all transactions | Search with transaction hash or address

```
decoded input         {
                              "address tokenOwner": "0x60f30Ed9926A27A74a0Ad5B11dAD4aa4Dbd53e29"
                      }
decoded output        {
                              "0": "uint256: balance 100000"
                      }
logs                  []

transact to QKCToken.approve errored: Error encoding arguments: Error: invalid BigNumber string (argument="value", value="", code=INVALID_ARGUMENT, version=bignumber/5.5.0)

transact to QKCToken.approve pending ...
```

MetaMask Notification

By granting permission, you are allowing the following account to access your funds

0x60f3...3e29

Edit Permission

Transaction fee Edit

A fee is associated with this request. $0.00
0.000072 RopstenETH

Nonce Edit 7

Hide full transaction details

Permission request Edit

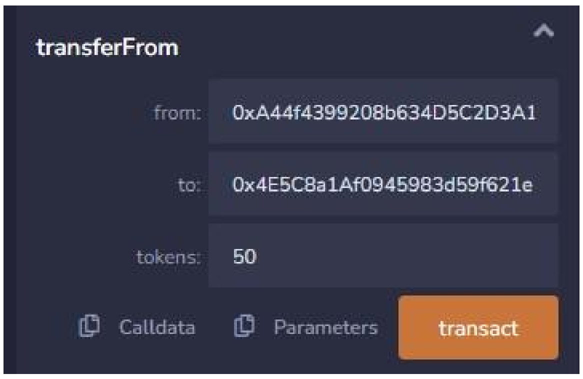
https://remix.ethereum.org may access and spend up to this max amount

Approved amount: 0.08 QKC

Granted to: 0x60f30Ed9...3e29

allowance: This method is exactly the same as the approved method except that it checks if one user has enough balance to send a certain amount of tokens to another.

OUTPUT :

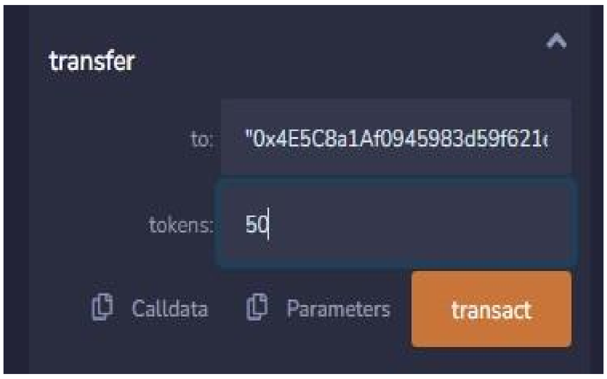


transferFrom

from: 0xA44f4399208b634D5C2D3A1

to: 0x4E5C8a1Af0945983d59f621e

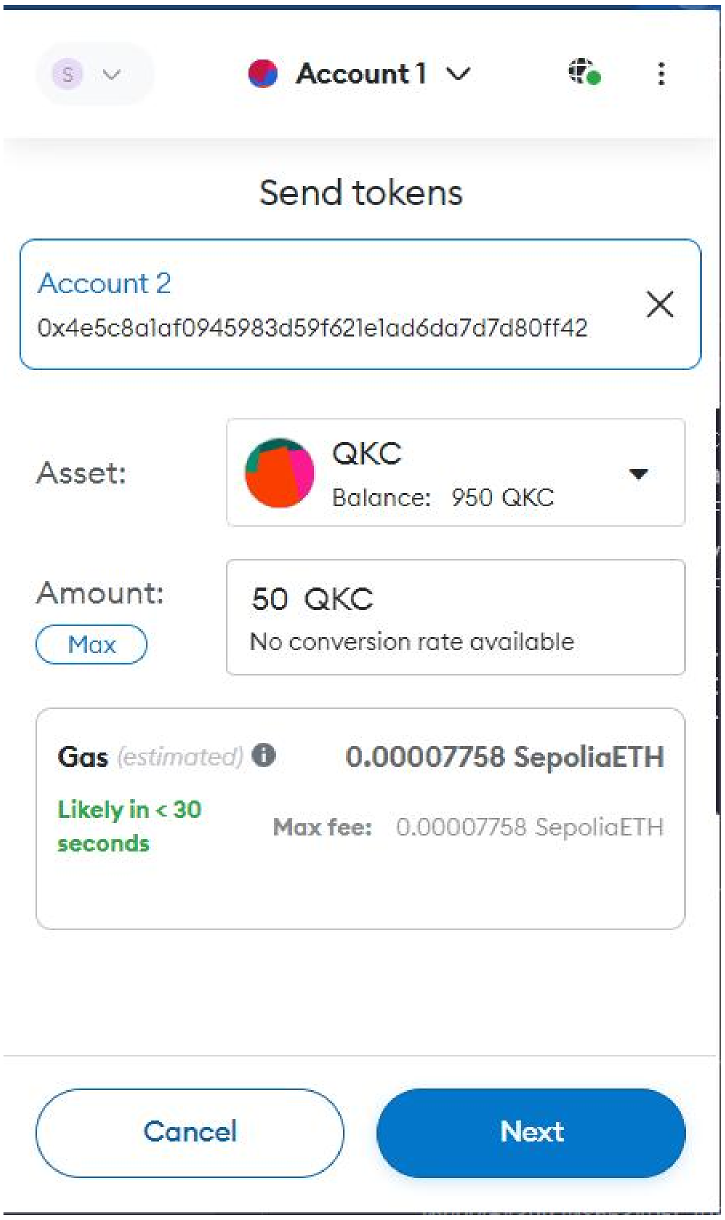
tokens: 50



transfer

to: "0x4E5C8a1Af0945983d59f621(

tokens: 50



Account 1

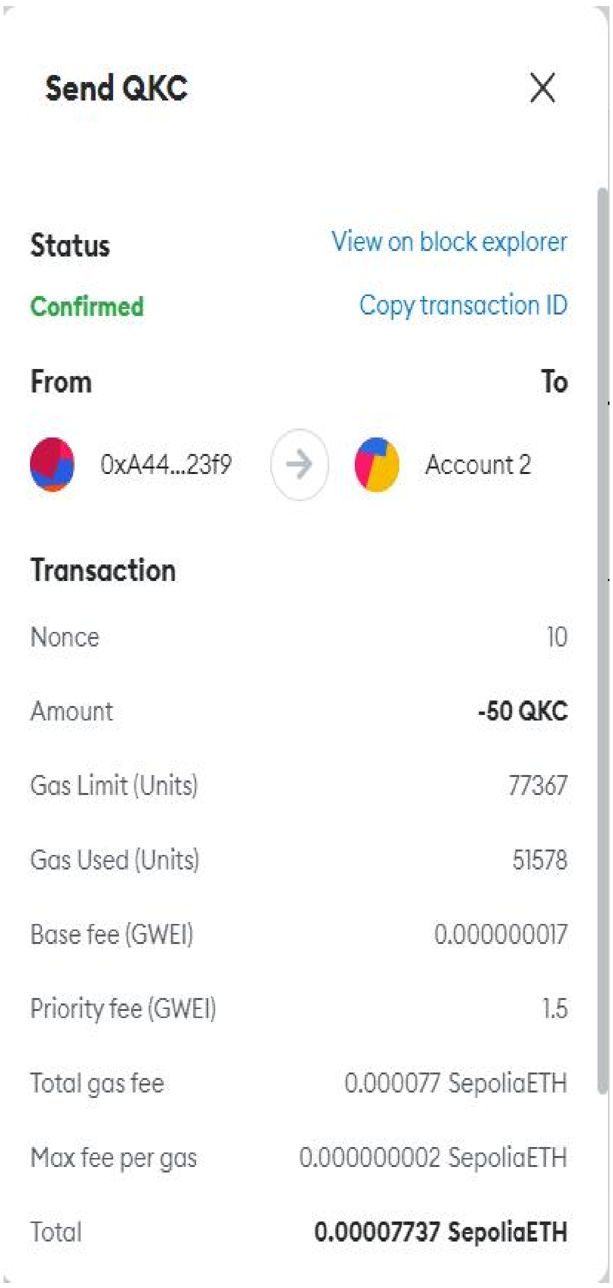
Send tokens

Account 2
0x4e5c8a1af0945983d59f621e1ad6da7d7d80ff42

Asset: QKC
Balance: 950 QKC

Amount: 50 QKC
No conversion rate available

Gas (estimated) 0.00007758 SepoliaETH
Likely in < 30 seconds
Max fee: 0.00007758 SepoliaETH

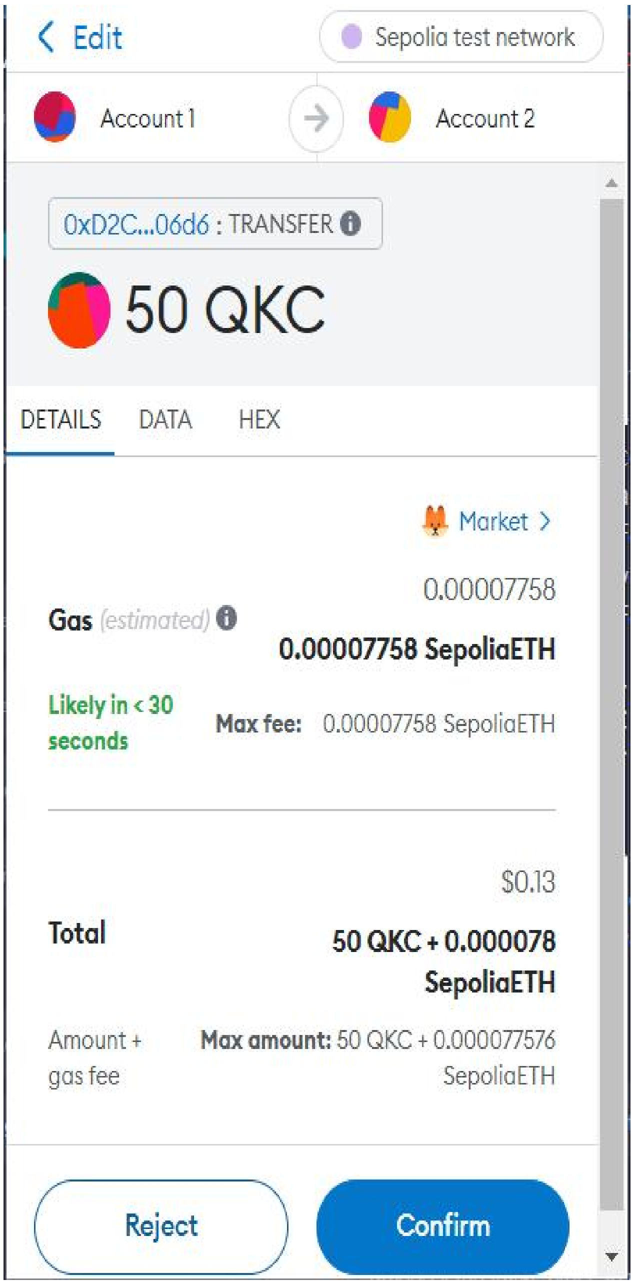


Send QKC

Status	View on block explorer
Confirmed	Copy transaction ID
From	**To**
0xA44...23f9	Account 2

Transaction

Nonce	10
Amount	**-50 QKC**
Gas Limit (Units)	77367
Gas Used (Units)	51578
Base fee (GWEI)	0.000000017
Priority fee (GWEI)	1.5
Total gas fee	0.000077 SepoliaETH
Max fee per gas	0.000000002 SepoliaETH
Total	**0.00007737 SepoliaETH**



Edit · Sepolia test network

Account 1 → Account 2

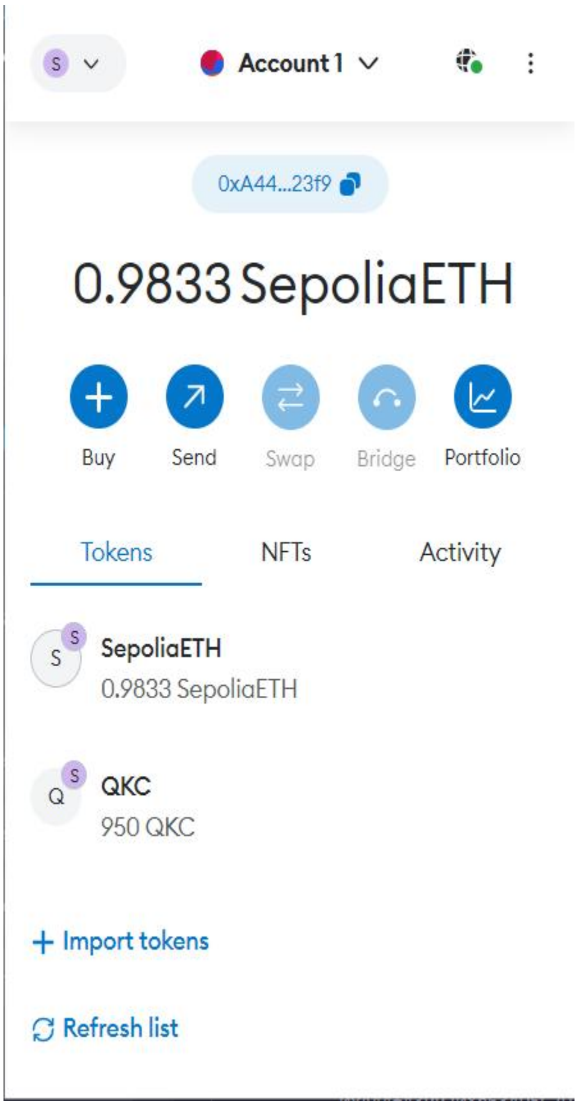
0xD2C...06d6 : TRANSFER

50 QKC

DETAILS DATA HEX

Market >

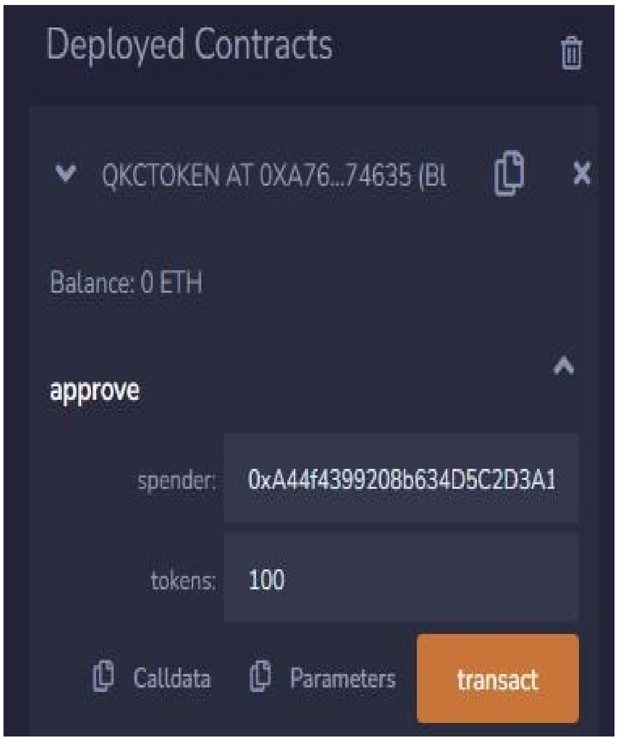
Gas *(estimated)*	0.00007758 **0.00007758 SepoliaETH**
Likely in < 30 seconds	**Max fee:** 0.00007758 SepoliaETH
Total	$0.13 **50 QKC + 0.000078 SepoliaETH**
Amount + gas fee	**Max amount:** 50 QKC + 0.00007576 SepoliaETH



0.9833 SepoliaETH

SepoliaETH
0.9833 SepoliaETH

QKC
950 QKC



QKCTOKEN AT 0XA76...74635 (Bl

Balance: 0 ETH

approve

spender: 0xA44f4399208b634D5C2D3A1

tokens: 100

Transferred 50 QKC from account 1 to 2

Note : Only 100 QKC can be approved at a time

Conclusion: We have succesfully created smart contract and embebeded wallet to perform the transactions.