

PRACTICAL: 3

AIM: Code Your Own Cryptocurrency on Ethereum (ERC-20 Token).

Course:

https://www.udemy.com/course/code-your-own-cryptocurrency/?srsltid=AfmBOopjvZsDJ6uJbYHew5nbOgyWiNRw0w2-x5fe6z_D53kyYdmjOQA3

THEORY:

ERC-20 is the technical standard for fungible tokens created using the Ethereum blockchain. A fungible token is one that is exchangeable with another token, whereas the well-known ERC-721 non-fungible tokens (NFTs) are not.

ERC-20 allows developers to create smart-contract-enabled tokens that can be used with other products and services. These tokens are a representation of an asset, right, ownership, access, cryptocurrency, or anything else that is not unique in and of itself but can be transferred.

Smart contracts were becoming more popular in 2015, but several issues needed to be addressed. One of the most pressing was that anyone could make a token, but they were not always interoperable with other tokens. Without a standardized token methodology, there wasn't a way to ensure that all the different tokens could be created, used, or exchanged by everyone using the blockchain.

ERC-20 was proposed by developer Fabian Vogelsteller in 2015 to address the need for a standard within smart contracts on the Ethereum blockchain. Vogelsteller submitted the proposal via the project's Github page as an Ethereum Request for Comment (ERC). As it was the twentieth comment, it was assigned the designation ERC-20.

Following the procedure used by the Ethereum developer community at the time, the proposal was approved and implemented in 2017 as Ethereum Improvement Proposal 20 (EIP-20). However, it is still called ERC-20 because that's how it was known until it was approved.

Because the request was approved and implemented, smart contract tokens implemented on the Ethereum blockchain must conform to this standard if the developers want them to be interchangeable and advertise that their token is ERC-20 compliant.

ERC-20 is a list of functions and events that must be implemented into a token to be considered ERC-20 compliant. These functions (called methods in the ERC) describe what must be included in the smart-contract-enabled token, while events describe an action. The functions a token must have been:

- **TotalSupply:** The total number of tokens that will ever be issued
- **BalanceOf:** The account balance of a token owner's account
- **Transfer:** Automatically executes transfers of a specified number of tokens to a specified address for transactions using the token

- **TransferFrom:** Automatically executes transfers of a specified number of tokens from a specified address using the token
- **Approve:** Allows a spender to withdraw a set number of tokens from a specified account, up to a specific amount
- **Allowance:** Returns a set number of tokens from a spender to the owner

The events that must be included in the token are:

- **Transfer:** An event triggered when a transfer is successful
- **Approval:** A log of an approved event (an event)

The following functions are optional and are not required, but they enhance the token's usability:

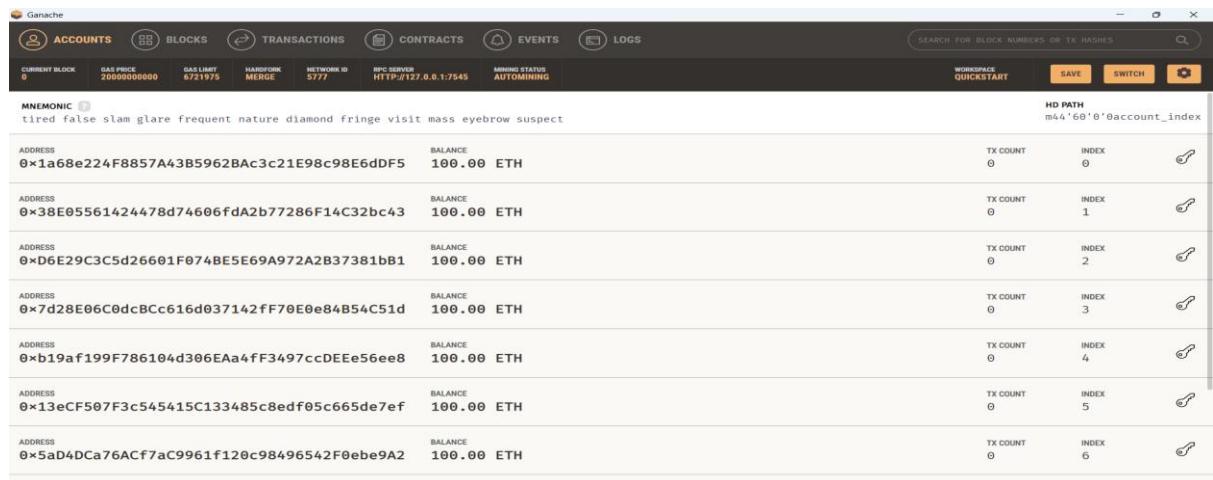
- Token's name (optional)
- Its symbol (optional)
- Decimal points to use (optional)

These functions provide a common structure for tokens so that they can be easily accessed, recognized, reviewed, and used. This reduces the confusion users and application developers would have if every smart contract's token had different information contained within it. Additionally, the code functions assist in determining the number of tokens in circulation, storing and returning balances, making transfer and withdrawal requests, granting approval, and agreeing to automated transfers.

CODE:

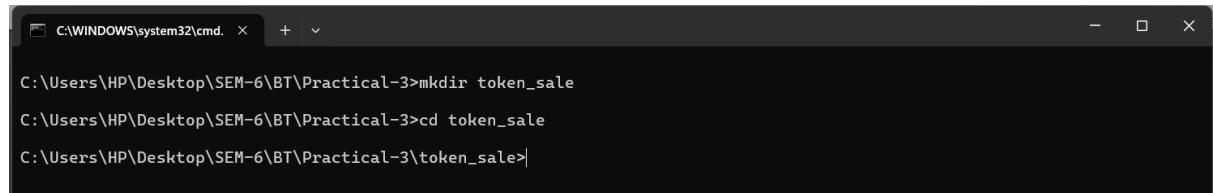
- npm install -g truffle
- npm install -g solc
- mkdir token_sale/
- cd token_sale
- truffle init
- echo.> contracts/Trushang.sol
- truffle migrate --reset
- truffle console
- npm install

OUTPUT:



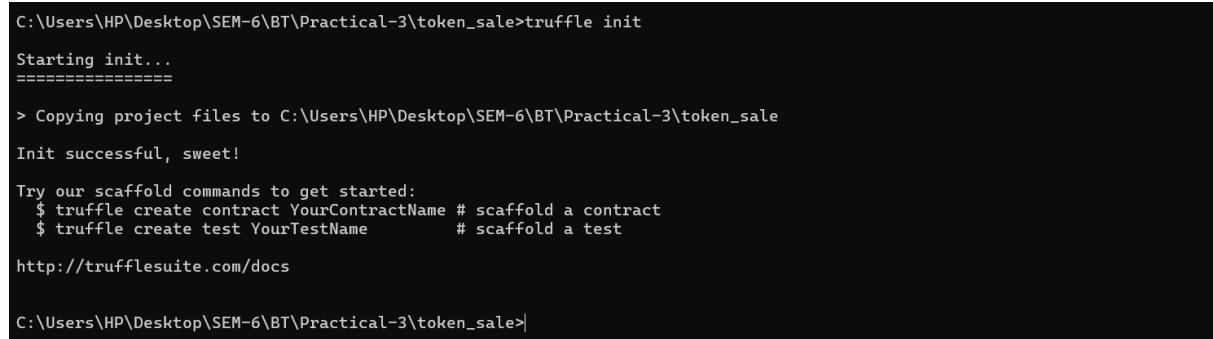
MNEMONIC	ADDRESS	BALANCE	TX COUNT	INDEX
tired false slam glare frequent nature diamond fringe visit mass eyebrow suspect	0x1a68e224f8857a43b5962Bac3c21E98c98E6dDF5	100.00 ETH	0	0
	0x38E05561424478d74606fdA2b77286F14C32bc43	100.00 ETH	0	1
	0xD6E29C3C5d26601F074BE5E69A972A2B37381bB1	100.00 ETH	0	2
	0x7d28E06C0dc8Cc616d037142ff70E0e84B54C51d	100.00 ETH	0	3
	0xb19af199F786104d306EAa4fF3497ccDEEe56ee8	100.00 ETH	0	4
	0x13eCF507F3c545415C133485c8edf05c665de7ef	100.00 ETH	0	5
	0xbuD4DCa76ACf7aC9961f120c98496542F0ebe9A2	100.00 ETH	0	6

Figure 1:Start Ganache



```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3>mkdir token_sale
C:\Users\HP\Desktop\SEM-6\BT\Practical-3>cd token_sale
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>
```

Figure 2:Create project directory



```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle init
Starting init...
=====
> Copying project files to C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale
Init successful, sweet!
Try our scaffold commands to get started:
$ truffle create contract YourContractName # scaffold a contract
$ truffle create test YourTestName           # scaffold a test
http://trufflesuite.com/docs
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>
```

Figure 3:create truffle project in directory

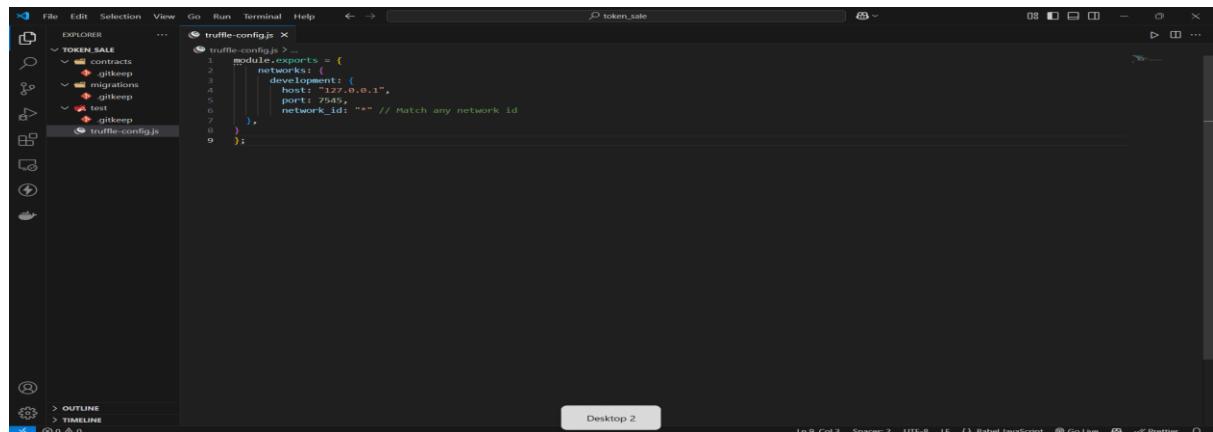


Figure 4:Config your truffle-config.js

```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale> echo > contracts/Trushang.sol
```

The terminal window shows the command 'echo > contracts/Trushang.sol' being run. The output is a blank file named 'Trushang.sol' in the 'contracts' directory.

Figure 5:Create a new Contract file

```
contracts > TrushangToken.sol > solidity > TrushangToken
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract TrushangToken{
5     //constructor
6     //Set the total number of tokens
7     //Read the total number of tokens
8     uint256 public totalSupply;
9     constructor(){
10         totalSupply = 1000000;
11     }
12 }
13 }
```

The code editor displays the 'TrushangToken.sol' file. It contains a single contract definition named 'TrushangToken'. The contract has a constructor that initializes the 'totalSupply' variable to 1,000,000.

Figure 6:Write your Token contract

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ token_sale
```

```
JS 1_deploy_contract.js X
migrations > JS 1_deploy_contract.js > exports
1 var TrushangToken = artifacts.require('contracts/TrushangToken.sol');
2
3 module.exports = function(deployer) {
4     deployer.deploy(TrushangToken);
5 }
```

The code editor shows the '1_deploy_contract.js' file in the 'migrations' directory. The script uses the Truffle framework to deploy the 'TrushangToken' contract.

Figure 7:Create a migration file

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ token_sale
```

```
JS 1_deploy_contract.js TrushangToken.sol truffle-config.js X
truffle-config.js > <unknown> > compilers > solc > version
1 module.exports = {
2   networks: {
3     development: {
4       host: "127.0.0.1",
5       port: 7545,
6       network_id: "*"
7     }
8   },
9   compilers: {
10     solc: [
11       {
12         version: "0.8.0"
13     }
14   ]
15 }
```

The code editor shows the 'truffle-config.js' file. It defines the network configuration for the 'development' environment and specifies the Solidity compiler version to be used for compilation.

Figure 8:write logic for migration and also update truffle-config.jS

```

PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle migrate --reset
Compiling your contracts...
=====
    ✓ Fetching sole version list from solc-bin. Attempt #1
    ✓ Downloading compiler. Attempt #1.
    > compiling \contracts\TrushangToken.sol
    > compiling \contracts\TrushangToken.sol
    > Artifacts written to: C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale\build\contracts
    > Compiled successfully using:
        - solc: 0.8.0+commit.cdf7d8e.emscripten clang

Starting migrations...
=====
> Network name: 'development'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)

1_deploy_contract.js
=====

Deploying 'TrushangToken'
=====
> transaction hash: 0x4c144cce130a2a65eddd24920d501775ac947b93935c922580d157a85391742
> Blocks: 0 Seconds: 0
> contract address: 0xd49e05e69948010d9898140b65e4e977b577c088
> block number: 1
> block timestamp: 1740663642
> account: 0x989c8119c17155326656525493f91b23f4c939d1
> balance: 99,999614139625
> gas used: 114329 (0x1be99)
> gas price: 3,375 gwei
> value sent: 0 ETH
> total cost: 0.000385860375 ETH

> Saving artifacts
=====
> Total cost: 0.000385860375 ETH

```

Figure 9:Run migration file

TX 0x4c144cce130a2a65eddd24920d501775ac947b93935c922580d157a85391742	
SENDER ADDRESS	0x989c8119c17155326656525493f91b23f4c939d1
CREATED CONTRACT ADDRESS	0xd49e05e69948010d9898140b65e4e977b577c088
VALUE	0.00 ETH
GAS USED	114329
GAS PRICE	3375000000
GAS LIMIT	142911
MINED IN BLOCK	1
TX DATA	
0x60806040523480156100105760080fd5b50620f4240600881905505060b3806100296000396000f3fe6080604052348015600f57600080fd5b506004361060285760003560e01c806318160ddd14602d575b600080fd5b60336047565b604051603e9190605a565b60405180910390f35b60005481565b6054816073565b82525050565b6000602082019050606d6000830184604d565b92915050565b600081905056fea264697067358221220693af79a2efeb7b600265c8398ba13f92d77a678ecf78d14f0d6e49bacc31f9b864736f6c63430000000033	

Figure 10:Check Ganache transaction

METAMASK

Account 7

99.9996 ETH

Ganache Ethereum

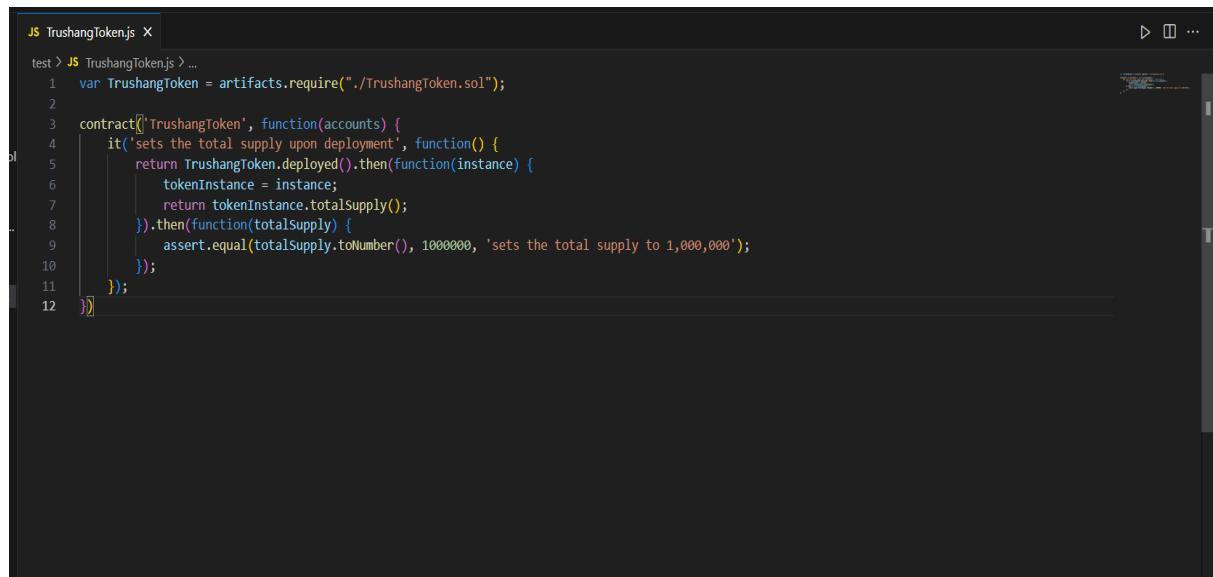
Buy & Sell Swap Bridge Send Receive

100 ETH

Figure 11:See Account in MetaMask

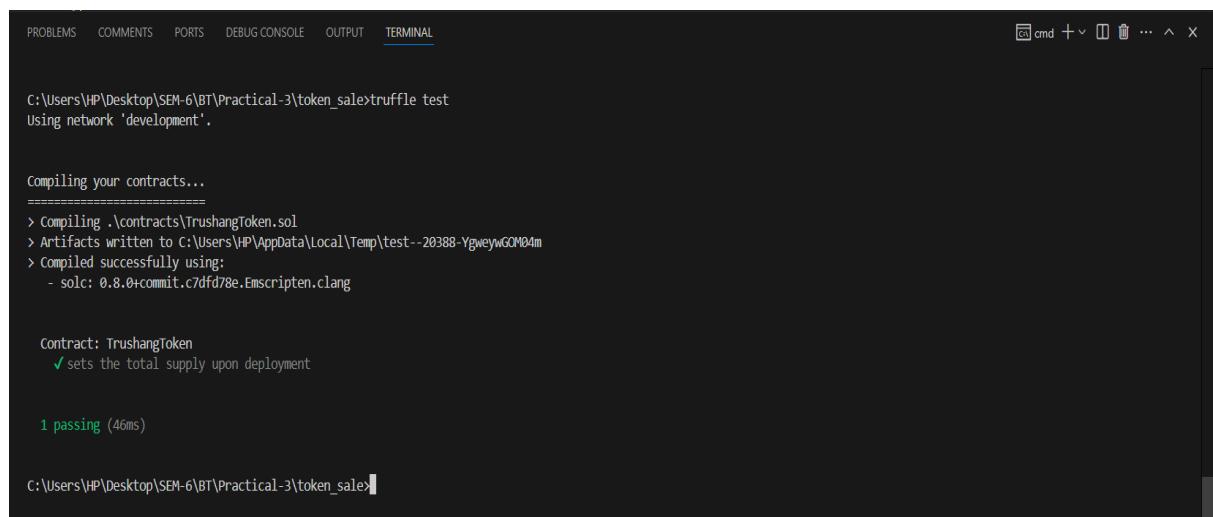
```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle console
truffle(development)> TrushangToken.deployed().then(function(i){token=i;})
undefined
truffle(development)> token.address
'0xd49e05e69948010d9898140b65E4E977b577c088'
truffle(development)> token.totalSupply().then(function(s){totalSupply=s;})
undefined
truffle(development)> totalSupply
BN {
  negative: 0,
  words: [ 1000000, <1 empty item> ],
  length: 1,
  red: null
}
truffle(development)> totalSupply.toNumber()
1000000
truffle(development)> .exit
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>
```

Figure 12: Open truffle console and verifying smart contract are created or not and if created then it's hash value and ganache contract value



```
JS TrushangToken.js X
test > JS TrushangToken.js > ...
1  var TrushangToken = artifacts.require("./TrushangToken.sol");
2
3  contract['TrushangToken', function(accounts) {
4    it('sets the total supply upon deployment', function() {
5      return TrushangToken.deployed().then(function(instance) {
6        tokenInstance = instance;
7        return tokenInstance.totalSupply();
8      }).then(function(totalSupply) {
9        assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 1,000,000');
10     });
11   });
12 }]
```

Figure 13: create test/TrushanToken.js file



```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle test
Using network 'development'.

Compiling your contracts...
=====
> Compiling ./contracts/TrushangToken.sol
> Artifacts written to C:/Users/HP/AppData/Local/Temp/test--20388-YgweywGOM04m
> Compiled successfully using:
  - solc: 0.8.0+commit.c7dfdf78e.Emscripten clang

Contract: TrushangToken
  ✓ sets the total supply upon deployment

1 passing (46ms)

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>
```

Figure 14: Testing the deployment

```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle test
Using network 'development'.

Compiling your contracts...
=====
> Compiling ./contracts\TrushangToken.sol
> Artifacts written to C:/Users/HP/appData/Local/Temp/test--7424-ROU3JuiquSbP
> Compiled successfully using:
  - solc: 0.8.0+commit.c7df78e.Emscripten clang

Contract: TrushangToken
  1) sets the total supply upon deployment
    > No events were emitted

  0 passing (47ms)
  1 failing

  1) Contract: TrushangToken
    sets the total supply upon deployment:
      sets the total supply to 1,000,000
        + expected - actual
        -1000000
        +8000000

at C:/Users/HP/Desktop\SEM-6\BT\Practical-3\token_sale\test\TrushangToken.js:9:20
at processTicksAndRejections (node:internal/process/task_queues:95:5)

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>
```

Figure 15:Change the value 1000000 to 800000 and again test it

```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>git init
Initialized empty Git repository in C:/Users/HP/Desktop\SEM-6\BT\Practical-3\token_sale/.git/
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    build/
    contracts/
    migrations/
    test/
    truffle-config.js

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>git add .

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>git commit -m "1. Somke Test"
[master (root-commit) 060fe2b] 1. Somke Test
 8 files changed, 478 insertions(+)
create mode 100644 build/contracts/trushangToken.json
create mode 100644 contracts/.gitkeep
create mode 100644 contracts/trushangToken.sol
create mode 100644 migrations/.gitkeep
create mode 100644 migrations/1_deploy_contract.js
create mode 100644 test/.gitkeep
create mode 100644 test/trushangToken.js
create mode 100644 truffle-config.js

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>git status
On branch master
nothing to commit, working tree clean

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>
```

Figure 16:Create git repo and add to the github

```
File Edit Selection View Go Run Terminal Help ⏎ → token_sale
EXPLORER TOKEN_SALE
  build
  contracts
    TrushangToken.sol
  migrations
    1_deploy_contract.js
  test
    TrushangToken.js
    truffle-config.js

1_deploy_contract.js M TrushangToken.sol M TrushangToken.js M
test > TrushangToken.js > contract('TrushangToken') callback > it('sets the total supply upon deployment') callback > then() callback
  1 var TrushangToken = artifacts.require("./TrushangToken.sol");
  2
  3 contract('TrushangToken', function(accounts) {
  4   var tokenInstance;
  5
  6   it('sets the total supply upon deployment', function() {
  7     return TrushangToken.deployed().then(function(instance) {
  8       tokenInstance = instance;
  9       return tokenInstance.totalSupply();
 10     }).then(function(totalSupply) {
 11       assert.equal(totalSupply.toNumber(), 1000000, 'sets the total supply to 1,000,000');
 12       return tokenInstance.balanceOf(accounts[0]);
 13     }).then(function(adminBalance) {
 14       assert.equal(adminBalance.toNumber(), 1000000, 'it allocates the initial supply to the admin account');
 15     });
 16   });
 17 })
```

Figure 17: Set initial value 1,000,000

```

File Edit Selection View Go Run Terminal Help <- > token_sale
EXPLORER TOKEN_SALE contracts TrushangToken.sol M TrushangToken.js
contracts > TrushangToken.sol ...
3
5 //constructor
6 //Set the total number of tokens
7 //Read the total number of tokens
8 uint256 public totalSupply;
9
10 mapping(address => uint256) public balanceOf;
11
12 constructor(uint256 _initialSupply){
13     balanceOf[msg.sender] = _initialSupply;
14     totalSupply = _initialSupply;
15     //allocate the initial supply
16 }
17
18
19 }
20

PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL
Compiling \Contracts\TrushangToken.sol
Compiling \Contracts\TrushangToken.sol
Artifacts written to C:\Users\HP\AppData\Local\Temp\test-14412-2GjjoYtXg54
Compiled successfully using:
- solc: 0.8.0+commit.c7dfd78e.Emscripten.clang

Contract: TrushangToken
✓ sets the total supply upon deployment

1 passing (53ms)

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale\]
Ln 1 Col 1 Spaces:4 UTF-8 CRLF Solidity Go Live Prettier

```

Figure 18:Test the code

```

PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle console
truffle(development)> web3.eth.getAccounts()
[
  '0x5704cc68C1f0ce03eb1fE0Ca25ad2910b0340f04',
  '0x4A10608029Af73E2Fb95dcB8Ef18fb912e62E304f',
  '0xCf823Ef111f5699A57064bb0fA156835f676899',
  '0x87A9945C0d906D92f11f172B85e2Ba283D2d17',
  '0xB95c506663045364e397985f593E5Dfcf14E3999',
  '0xd7d28abd74Dc24170eEdE5072fAAA2014268a5',
  '0xeFcfa44e059680f09AcF30d2f2452eb7d08346e5',
  '0x4e9f23853928A51Abd5d40Af054678C913C253',
  '0x0f563d0f65ab50520Ff0b6980a026a585F410cD',
  '0xE97a580fb1676CC7213B2A00b2397133B220a7a5'
]
truffle(development)> 

```

Figure 19:In truffle see how many accounts in our network using web3.eth.getAccounts()

```

File Edit Selection View Go Run Terminal Help <- > token_sale
JS 1_deploy_contract.js M TrushangToken.sol M JS TrushangToken.js M
contracts > TrushangToken.sol > solidity > TrushangToken standard
4
5 pragma solidity ^0.8.0;
6
7 contract TrushangToken{
8     //constructor
9     //Set the total number of tokens
10    //Read the total number of tokens
11
12    //Name
13    string public name = "Trushang Token";
14    //Symbol
15    string public symbol = "TRU";
16    //Standard
17    string public standard = "Trushang Token v1.0";
18
19    uint256 public totalSupply;
20
21    mapping(address => uint256) public balanceOf;
22
23    constructor(uint256 _initialSupply){
24        balanceOf[msg.sender] = _initialSupply;
25        totalSupply = _initialSupply;
26        //allocate the initial supply
27    }
28
29
30 }

PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL
> Compiled successfully using:
- solc: 0.8.0+commit.c7dfd78e.Emscripten.clang

Contract: TrushangToken
✓ initializes the contract with the correct values (45ms)
✓ allocates the initial supply upon deployment

2 passing (117ms)

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale\]
Ln 1 Col 1 Spaces:4 UTF-8 CRLF Solidity Go Live Prettier

```

Figure 20:Add token name, symbol, and standard in our code and then test it

```

JS 1_deploy_contract.js      TrushangToken.sol M      JS TrushangToken.js M
contracts > TrushangToken.sol > solidity > TrushangToken
13
14     event Transfer(
15         address indexed _from,
16         address indexed _to,
17         uint256 _value
18     );
19
20     mapping(address => uint256) public balanceOf;
21
22     //Constructor
23     //Set the total number of tokens
24     //Read the total number of tokens
25     constructor(uint256 _initialSupply) {
26     }
27
28     //Transfer
29     function transfer(address _to, uint256 _value) public returns(bool success) {
30         // Check for valid address
31         require(_to != address(0), "Invalid recipient address");
32
33         // Check if sender has enough balance with revert message
34         require(balanceOf[msg.sender] >= _value, "insufficient balance, transfer rejected");
35
36         // Transfer tokens
37         balanceOf[msg.sender] -= _value;
38         balanceOf[_to] += _value;
39
40         // Emit transfer event
41         emit Transfer(msg.sender, _to, _value);
42
43         return true;
44     }
45
46 }
47
48

```

Figure 21: Create Construction

```

33     it('transfers token ownership', function() {
34         return trushangToken.deployed().then(function(instance) {
35             tokenInstance = instance;
36             // Try transferring something larger than the sender's balance
37             return tokenInstance.transfer.call(accounts[1], 9999999999999999);
38         }).then(assert.fail).catch(function(error) {
39             assert(
40                 error.message.indexOf('revert') >= 0,
41                 'should revert on insufficient balance'
42             );
43             return tokenInstance.transfer(accounts[1], 250000, { from: accounts[0] });
44         }).then(function(receipt) {
45             assert.equal(receipt.logs.length, 1, 'triggers one event');
46             assert.equal(receipt.logs[0].event, 'Transfer', 'should be the "Transfer" event');
47             assert.equal(receipt.logs[0].args_.from, accounts[0], 'logs the account the tokens are transferred from');
48             assert.equal(receipt.logs[0].args_.to, accounts[1], 'logs the account the tokens are transferred to');
49             assert.equal(receipt.logs[0].args_.value, 250000, 'logs the transfer amount');
50             return tokenInstance.balanceOf(accounts[1]);
51         }).then(function(balance) {
52             assert.equal(balance.toNumber(), 250000, 'adds the amount to the receiving account');
53             return tokenInstance.balanceOf(accounts[0]);
54         }).then(function(balance) {
55             assert.equal(balance.toNumber(), 750000, 'deducts the amount from the sending account');
56         });
57     });
58 });

```

Figure 22: Create a Token Ownership

```

PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL
G:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle test
Using network 'development'.

Compiling your contracts...
=====
> Compiling ./contracts/TrushangToken.sol
> Compiling ./contracts/TrushangToken.sol
Artifacts written to C:\Users\HP\AppData\Local\Temp\test--17348-L0022rkC15eZ
Compiled successfully using:
- solc: 0.8.0+commit.c7dfd78c.Emscripten.clang

Contract: TrushangToken
Contract: TrushangToken
    ✓ initializes the contract with the correct values (40ms)
    ✓ initializes the contract with the correct values (40ms)
    ✓ allocates the initial supply upon deployment
    ✓ transfers token ownership (287ms)
    ✓ transfers token ownership (287ms)

3 passing (396ms)
3 passing (396ms)

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>

```

Figure 23: Test the contract

The screenshot shows the Ganache interface with several transaction logs listed:

- TX HASH: 0xa2163807285af122b54c88df6a69923f0b7da7ed11e81ee64ca26df5ab471ae4**
From Address: 0x5704cc68C1f6Ce03eb1fE0Ca25adE29100340f04
To Contract Address: 0xd5FFd3b1C82B0C76524a0b85B6eFdF8B0b6159E5
Gas Used: 52434
Value: 0
- TX HASH: 0x26fd84fa25ccd5b9736be297df08dc2d631e43a75aa9492c027ca00c288e8e15**
From Address: 0x5704cc68C1f6Ce03eb1fE0Ca25adE29100340f04
Created Contract Address: 0x55809487bAFFD9Fc2ebD5dd01DE23b1e7e4Ba3A
Gas Used: 691917
Value: 0
- TX HASH: 0x3af0dddac78e629cca6e540ce54ed2146cdc0326f522b715e2c369c73946c268**
From Address: 0x5704cc68C1f6Ce03eb1fE0Ca25adE29100340f04
Created Contract Address: 0xD4a92A547ED8C11267918c00016D1ab43FF15281
Gas Used: 587650
Value: 0
- TX HASH: 0xe167bd79898fc1fe03c227e69be73e57291de25fe30a6ec67cf07ea10372f301**
From Address: 0x5704cc68C1f6Ce03eb1fE0Ca25adE29100340f04
Created Contract Address: 0x55809487bAFFD9Fc2ebD5dd01DE23b1e7e4Ba3A
Gas Used: 585828
Value: 0
- TX HASH: 0x994cb55a503c7e96bb9d623ff65344bbd6b5277cc8f97e5fd79ef6abb8f14d0b**
From Address: 0x5704cc68C1f6Ce03eb1fE0Ca25adE29100340f04
Created Contract Address: 0x55809487bAFFD9Fc2ebD5dd01DE23b1e7e4Ba3A
Gas Used: 500000
Value: 0

Figure 24: Ganach transaction for ownership transfer

```

JS 1_deploy_contract.js  TrushangToken.sol M  JS TrushangToken.js M
contracts > TrushangToken.sol > solidity > TrushangToken
3
22 //allowance
23
24 //approve
25 event Approval(
26     address indexed _owner,
27     address indexed _spender,
28     uint256 _value
29 );
30
31 //Constructor
32 > constructor(uint256 _initialSupply){
33 }
34
35 //Transfer
36 > function transfer(address _to, uint256 _value) public returns(bool success) {
37 }
38
39 //Approve
40 > function approve(address _spender, uint256 _value) public returns(bool success){
41     //allowance
42     //Approval event
43     allowance[msg.sender][_spender] = _value;
44     emit Approval(msg.sender, _spender, _value);
45
46     return true;
47 }
48
49 }

```

Figure 25: Create approve contract

```

59
60     it('approves tokens for delegated transfer', function() {
61         return TrushangToken.deployed().then(function(instance) {
62             tokenInstance = instance;
63             return tokenInstance.approve.call(accounts[1], 100);
64         }).then(function(success) {
65             assert.equal(success, true, 'it returns true');
66             return tokenInstance.approve(accounts[1], 100, { from: accounts[0] });
67         }).then(function(receipt) {
68             assert.equal(receipt.logs.length, 1, 'triggers one event');
69             assert.equal(receipt.logs[0].event, 'Approval', 'should be the "Approval" event');
70             assert.equal(receipt.logs[0].args._owner, accounts[0], 'logs the account the tokens are authorized by');
71             assert.equal(receipt.logs[0].args._spender, accounts[1], 'logs the account the tokens are authorized to');
72             assert.equal(receipt.logs[0].args._value, 100, 'logs the transfer amount');
73             return tokenInstance.allowance(accounts[0], accounts[1]);
74         }).then(function(allowance) {
75             assert.equal(allowance.toNumber(), 100, 'stores the allowance for delegated transfer');
76         });
77     });
78 });
79

```

Figure 26: Write Approves tokens for delegated transfer function

Figure 27: Test Approve token

Blockchain Status							Search Bar		
Accounts	Blocks	Transactions	Contracts	Events	Logs	Search for Block Numbers or Tx Hashes			
CURRENT BLOCK 18	GAS PRICE 2000000000	GAS LIMIT 6721975	HARDFORK MERGE	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING	WORKSPACE QUICKSTART	SAVE	SWITCH
TX HASH 0x9f952b5601c2b3bf3c2cab6c32b0f96b955251aa2467ab7f2f4ab0ec806f3993	FROM ADDRESS 0x3CB2cD74867d76f3FFe6f30E758822ca740c41e	TO CONTRACT ADDRESS 0x6502DFE6a03ff449BA8413924C1Fd454592573DE	GAS USED 46670	VALUE 0	CONTRACT CALL				
TX HASH 0x315cb53322aa794d53af9d9795e2b34780f8b8d475c654279e663aaad38fab03	FROM ADDRESS 0x33CB2cD74867d76f3FFe6f30E758822ca740c41e	TO CONTRACT ADDRESS 0x6502DFE6a03ff449BA8413924C1Fd454592573DE	GAS USED 52398	VALUE 0	CONTRACT CALL				
TX HASH 0xf281da35e912f5cc81ec42ecb46640fc0fbfe60e1b369d5b37af268e0e13449a	FROM ADDRESS 0x33CB2cD74867d76f3FFe6f30E758822ca740c41e	CREATED CONTRACT ADDRESS 0x6502DFE6a03ff449BA8413924C1Fd454592573DE	GAS USED 794024	VALUE 0	CONTRACT CREATION				
TX HASH 0xbc20c862024b62b75e153f3b8a1d4522870f7fadf861e9bb62d1c072b29777ca	FROM ADDRESS 0x33CB2cD74867d76f3FFe6f30E758822ca740c41e	TO CONTRACT ADDRESS 0x0D21b8F14F5ea715AeF0100BA90Add1b49473b1	GAS USED 46670	VALUE 0	CONTRACT CALL				
TX HASH 0x9fd6d6457a6ceaa366dee32f3d4f92c2936fe9e40aa2bb89f560975f8ac8f914	FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE	CONTRACT CALL				

Figure 28:Approved token transfer in Ganache

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
139
140
141
142
143
144
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
218
219
219
220
221
222
223
224
225
226
227
228
229
229
230
231
232
233
234
235
236
237
237
238
239
239
240
241
242
243
244
245
245
246
247
247
248
249
249
250
251
252
253
254
255
255
256
257
257
258
259
259
260
261
261
262
263
263
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
```

Figure 29: Write smart contract for transfer

Figure 30: Transfer function

Ganache											
ACCOUNTS		BLOCKS		TRANSACTIONS		CONTRACTS		EVENTS			
CURRENT BLOCK 24	GAS PRICE 2000000000	GAS LIMIT 6721975	HARDFORK MERGE	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING		LOGS			
MNEMONIC 🌐					appear time amused cage skirt rival fly twin river fury chase more						
					HD PATH m/44'60'0*0account_index						
ADDRESS 0x33CB2cD74867d76f3FFe6f30E758822ca740c41e	BALANCE 99.98 ETH		TX COUNT 22		INDEX 0		🔗				
ADDRESS 0x571606a645CABD043cEfC2b344b8148CD505b265	BALANCE 100.00 ETH		TX COUNT 0		INDEX 1		🔗				
ADDRESS 0xEb6CB089955A490Ef70891385bC9E46E87599e02	BALANCE 100.00 ETH		TX COUNT 1		INDEX 2		🔗				
ADDRESS 0xb9D5410aF2Ee83b758416664d7ef347bc1CAc32f	BALANCE 100.00 ETH		TX COUNT 0		INDEX 3		🔗				
ADDRESS 0xfAFF1cDb583C4AF0D725F065E7822e4242094A8D	BALANCE 100.00 ETH		TX COUNT 1		INDEX 4		🔗				
ADDRESS 0xBc36564Aea2917d99942cc10a1e0b0D35851cd34	BALANCE 100.00 ETH		TX COUNT 0		INDEX 5		🔗				
ADDRESS 0x3294B40D2b120e3e81239841199557960E4E0d43	BALANCE 100.00 ETH		TX COUNT 0		INDEX 6		🔗				

Figure 31: Ganache transaction

Figure 32:Single Ganache transaction

```
PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL

C:\Users\VP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle test
Using network "development".

Compiling your contracts...
> Compiling ./contracts/TrushangToken.sol
> Artifacts written to C:\Users\VP\AppData\Local\Temp\test--4844-DGdM6RoaJXO
> Compiled successfully using:
- solc: 0.8.0+commit.c7ff07be.Emscripten.clang

Contract: TrushangToken
✓ deploys a new contract with the correct values (50ms)
✓ allocates the initial supply upon deployment
✓ transfers token ownership (20ms)
✓ approves tokens for delegated transfer (82ms)
✓ handles delegated token transfers (560ms)

5 passing (978ms)

C:\Users\VP\Desktop\SEM-6\BT\Practical-3\token_sale>
```

Figure 33: Test Delegated Token Transfers

```
File Edit Selection View Go Run Terminal Help < > token_sale
PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL
cmd + x

> Compiling .\contracts\TrushangToken.sol
> Artifacts written to C:\Users\VIP\Desktop\SEM-6\BT\Practical-3\token_sale\build\contracts
> Compiled successfully using:
- solc: 0.8.0+commit.c7df98e.Emscripten clang

starting migrations...
=====
> Network name: 'development'
> Network id: 5777
> Block gas limit: 672975 (0x6691b7)

1_deploy_contract.js
=====
Replacing 'TrushangToken'
=====
> transaction hash: 0xb6d2b23d9efcad9d828273dab21eb6d9f4c97144edb381e322aa1aa15ad
> block: 0 Seconds: 0
> contract address: 0xC201C1397670aC623d46625B81838C2004010b
> block number: 31
> block timestamp: 1740688162
> account: 0x3C92cD74867d76f3F6e6f30E758822ca740c41e
> balance: 99.976897225217403826
> gas used: 1014610 (0xf7hb52)
> gas price: 2.521846248 gwei
> value sent: 0 ETH
> total cost: 0.00255869042168328 ETH

> Saving artifacts
=====
> Total cost: 0.00255869042168328 ETH

Summary
=====
> Total deployments: 1
> Final cost: 0.00255869042168328 ETH
```

Figure 34:Migrate the contract

```
PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL

node +v 🌐 🌐 🌐 ... ^

C:\Users\HP\Desktop\SEM-6-BT\Practical-3\token_sale>truffle console
truffle(development)> TrushangToken.deployed().then(function(instance){tokenId=instance;})
undefined
truffle(development)> tokenId.name()
'Trushang Token'
truffle(development)> tokenId.symbol()
'TRU'
truffle(development)> tokenId.standard()
'Trushang Token v1.0'
truffle(development)> tokenId.address
'0xCC201c1C3976704e623d46625801838c2B04010b'
truffle(development)> accounts
[
  '0x3C82xD748b57d76f3Fe6f30E75882c740c41e',
  '0x571606da45c5d0043cEfc2b34d8148C0505b265',
  '0xbcb2B8959545490eF78891385bc9e46e87599e02',
  '0xb905410af2ce3b75841664d7f37b1cAc3f7',
  '0xfAf1c0583cAAf80725f0657822e4242994a80',
  '0x1c85644ea291799942c10a1e0b0035851c034',
  '0x32948002129e3881239841199557968e4E0243',
  '0xA725A7C558668391dA063654141c7f1648fEc94',
  '0xfcfe657581881BF07f55471b2d28449f63fA6339',
  '0x7Alcaf06159acb99f1e0110c0aa8b35c41b299F'
]
truffle(development)> ■
```

Figure 35:Check accounts in our Blockchain

Figure 36: Transfer token from adim to account[1]

Ganache

ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK 32 GAS PRICE 2000000000 GAS LIMIT 6721975 HARDFORK MERGE NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING

WORKSPACE QUICKSTART SAVE SWITCH

STORAGE

```
↓ ( 6 items
  name : string "Trushang Token"
  symbol : string "TRU"
  standard : string "Trushang Token v1.0"
  totalSupply : uint 154240
  ▶ balanceOf : () mapping 0 items
  ▶ allowance : () mapping 0 items
}
```

TRANSACTIONS

TX HASH	FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE	CONTRACT CALL
0xb3482b31a30a16fb84c4237bf9a92ab492c0730199176d9604e2d88a446c3770	0x33CB2cD74867d76f3FFe6f30E758822ca740c4le	TrushangToken	52388	0	

EVENTS

EVENT NAME	TX HASH	LOG INDEX	BLOCK TIME
Transfer	0xb3482b31a30a16fb84c4237bf9a92ab492c0730199176d9604e2d88a446c3770	0	2025-02-28 08:06:31

Figure 37: See transaction Ganache

Figure 38:Approve transaction

```
    Rec: null
}
truffle(development)> fromAccount = accounts[2];
'0xEb6CB089955A490Ef70891385bc9E46E87599e02'
truffle(development)> toAccount = accounts[3];
truffle(development)> tokenInstance.transfer(fromAccount,100,{from:accounts[0]})
truffle(development)> tokenInstance.transfer(fromAccount,100,{from:accounts[0]})
{
truffle(development)> tokenInstance.transfer(fromAccount,100,{from:accounts[0]})
{
```

Figure 39: Transfer token

Figure 40:Transfer the token

Ganache

ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK 35 GAS PRICE 2000000000 GAS LIMIT 6721975 HARDFORK MERGE NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING

WORKSPACE QUICKSTART SAVE SWITCH

BACK 0x43cef79ce77922b3a5ae488756e8638dc56a44dbf4e8fb510b9367a111520200 (0)

CONTRACT NAME TrushangToken

CONTRACT ADDRESS 0xCC201c1C397670aC623d46625B81838C2B04010b

SIGNATURE (DECODED)
Approval(_owner: address, _spender: address, _value: uint256)

TX HASH 0x43cef79ce77922b3a5ae488756e8638dc56a44dbf4e8fb510b9367a111520200 LOG INDEX 0 BLOCK TIME 2025-02-28 00:27:04

RETURN VALUES

_OWNER
0xeb6cb089955a490ef70891385bc9e46e87599e02

_SPENDER
0xfaaff1cdb583c4af0d725f065e7822e4242094a8d

_VALUE
10

Figure 41: Transfer token transaction

```
}
```

```
truffle(development)> tokenInstance.balanceOf(fromAccount)
BN { negative: 0, words: [ 90, <1 empty item> ], length: 1, red: null }


```
truffle(development)> tokenInstance.balanceOf(toAccount)
BN { negative: 0, words: [10, <1 empty item>], length: 1, red: null }


```
truffle(development)> tokenInstance.allowance(fromAccount,spendingaccount);
Uncaught ReferenceError: spendingaccount is not defined

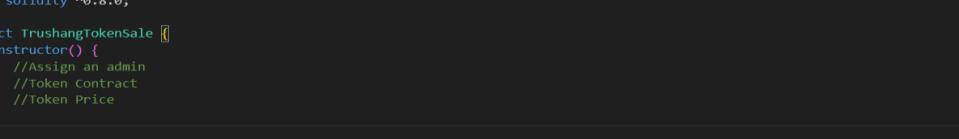
```


```


```

Figure 42: See the Balance of account

Figure 43: Transferform function transaction



The screenshot shows a terminal window with three tabs at the top: 'TrushangTokenSale.sol' (U), 'TrushangTokenSale.js' (S), and '1_deploy_contract.js' (M). The main area displays the Solidity code for 'TrushangTokenSale'. The code includes a SPDX license header, a pragma statement, and a constructor function that assigns an admin, creates a token contract, and sets a token price.

```
contracts > TrushangTokenSale.sol > solidity > TrushangTokenSale
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract TrushangTokenSale {
5     constructor() {
6         //Assign an admin
7         //Token contract
8         //Token Price
9     }
10 }
```

Figure 44:Start Tokenseale contract

```

test > JS TrushangTokenSale.js > ✘ contract('TrushangTokenSale') callback > ✘ it('initializes the contract with the correct values') callback > ✘ then() callback
  1 var TrushangTokenSale = artifacts.require('./TrushangToken.sol');
  2
  3 contract('TrushangTokenSale', function(accounts) {
  4   var tokenSaleInstance;
  5
  6   it('initializes the contract with the correct values', function() {
  7     return TrushangTokenSale.deployed().then(function(instance) {
  8       tokenSaleInstance = instance;
  9       return tokenSaleInstance.address;
 10     }).then(function(address) {
 11       assert.notEqual(address, 0x0, 'has contract address');
 12     });
 13   });
 14 });

```

Figure 45: Write Tokensale contract

```

PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle test
Using network 'development'.

Compiling your contracts...
=====
> Compiling .\contracts\TrushangToken.sol
> Compiling .\contracts\TrushangTokenSale.sol
> Artifacts written to C:\Users\HP\AppData\Local\Temp\test--21312-D0aSPtHgpOGF
> Compiled successfully using:
  - solc: 0.8.0+commit.c7dfd78e.Emscripten clang

Contract: TrushangToken
  ✓ initializes the contract with the correct values
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (231ms)
  ✓ approves tokens for delegated transfer (73ms)
  ✓ handles delegated token transfers (308ms)

Contract: TrushangTokenSale
  ✓ initializes the contract with the correct values

6 passing (756ms)

```

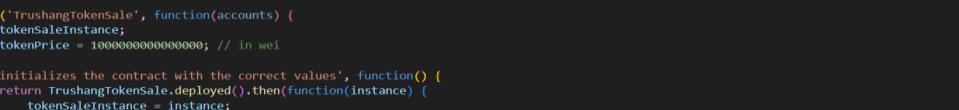
Figure 46: Test contract

```

migrations > JS 1_deploy_contract.js ...
● 1 var TrushangToken = artifacts.require('./TrushangToken.sol');
2 | var TrushangTokenSale = artifacts.require('./TrushangTokenSale.sol');
3
4 v module.exports = function(deployer) {
5   deployer.deploy(TrushangToken, 1000000).then(function() {
6     // Token price is 0.001 Ether
7     var tokenPrice = 1000000000000000;
8     return deployer.deploy(TrushangTokenSale, TrushangToken.address, tokenPrice);
9   });
10 };
11

```

Figure 47: Add tokensale in deployment contract



```
TrushangTokenSale.sol JS TrushangTokenSale.js M 1. deploy_contract.js M test > JS TrushangTokenSale.js > ...
1 var TrushangTokenSale = artifacts.require('./TrushangTokenSale.sol');
2
3 contract('TrushangTokenSale', function(accounts) {
4   var tokenSaleInstance;
5   var tokenPrice = 1000000000000000; // in wei
6
7     it('initializes the contract with the correct values', function() {
8       return TrushangTokenSale.deployed().then(function(instance) {
9         tokenSaleInstance = instance;
10        return tokenSaleInstance.address;
11      }).then(function(address) {
12        assert.notEqual(address, 0x0, 'has contract address');
13        return tokenSaleInstance.tokenContract();
14      }).then(function(address) {
15        assert.notEqual(address, 0x0, 'has token contract address');
16        return tokenSaleInstance.tokenPrice();
17      }).then(function(price) {
18        assert.equal(price,tokenPrice, 'token price is correct');
19      });
20    });
21  });

```

Figure 48: Write tokensale function



The screenshot shows a browser-based interface for writing Solidity smart contracts. The top navigation bar includes tabs for 'TrushangTokenSale.sol' (selected), 'TrushangTokenSale.js', and 'deploy_contract.js'. The main area displays the following Solidity code:

```
contracts > TrushangTokenSale.sol > solidity > TrushangTokenSale
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 import "./TrushangToken.sol";
5
6 contract TrushangTokenSale {
7     address admin;
8     TrushangToken public tokenContract;
9     uint256 public tokenPrice;
10
11    constructor(TrushangToken _tokenContract,uint256 _tokenPrice) {
12        //Assign an admin
13        admin = msg.sender;
14        //Token Contract
15        tokenContract = _tokenContract;
16        //Token Price
17        tokenPrice = _tokenPrice;
18
19    }
20 }
```

Figure 49: Write tokensale contract

```
PROBLEMS COMMENTS PORTS DEBUG CONSOLE OUTPUT TERMINAL

cmd + x ☰ ... ×

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle test
Using network 'development'.

Compiling your contracts...
> Compiling ./contracts\TrushangToken.sol
> Compiling ./contracts\TrushangTokenSale.sol
> Artifacts written to C:\Users\HP\AppData\Local\Temp\test--17000-oeDvxmRy36RX
> Compiled successfully using:
- solc: 0.8.0+commit.cdf7d78e.Emscripten clang
> Compiling ./contracts\TrushangToken.sol
> Compiling ./contracts\TrushangTokenSale.sol
> Artifacts written to C:\Users\HP\AppData\Local\Temp\test--17000-oeDvxmRy36RX
> Compiled successfully using:
- solc: 0.8.0+commit.cdf7d78e.Emscripten clang
> Artifacts written to C:\Users\HP\AppData\Local\Temp\test--17000-oeDvxmRy36RX
> Compiled successfully using:
- solc: 0.8.0+commit.cdf7d78e.Emscripten clang
> Artifacts written to C:\Users\HP\AppData\Local\Temp\test--17000-oeDvxmRy36RX
> Compiled successfully using:
- solc: 0.8.0+commit.cdf7d78e.Emscripten clang
> Compiled successfully using:
- solc: 0.8.0+commit.cdf7d78e.Emscripten clang

Contract: TrushangToken
Contract: TrushangToken
✓ initializes the contract with the correct values
✓ initializes the contract with the correct values
✓ allocates the initial supply upon deployment
✓ allocates the initial supply upon deployment
✓ transfers token ownership (174ms)
✓ transfers token ownership (174ms)
✓ approves tokens for delegated transfer (55ms)
✓ handles delegated token transfers (222ms)

Contract: TrushangTokenSale
✓ approves tokens for delegated transfer (55ms)
✓ handles delegated token transfers (222ms)

Contract: TrushangTokenSale
Ln 17 Col 34 Spaces: 4 UTF-8 CRLF ⓘ Solidity ⓘ Go Live ⓘ Prettier ⓘ
```

Figure 50:Test the contract



```

27
28   it('facilitates token buying', function() {
29     return TrushangToken.deployed().then(function(instance) {
30       //Grab token instance first
31       tokenInstance = instance;
32       return TrushangTokensale.deployed();
33     }).then(function(instance) {
34       //Then grab token sale instance;
35       tokenSaleInstance = instance;
36       // Provision 75% of all tokens to the token sale
37       return tokenInstance.transfer(tokenSaleInstance.address, tokensAvailable, { from: admin });
38     }).then(function(receipt) {
39       numberoftokens = 10;
40       return tokenSaleInstance.buyTokens(numberoftokens, { from: buyer, value: numberoftokens * tokenPrice });
41     }).then(function(receipt) {
42       assert.equal(receipt.logs.length, 1, 'triggers one event');
43       assert.equal(receipt.logs[0].event, 'Sell', 'should be the "Sell" event');
44       assert.equal(receipt.logs[0].args._buyer, buyer, 'logs the account that purchased the tokens');
45       assert.equal(receipt.logs[0].args._amount, numberoftokens, 'logs the number of tokens purchased');
46       return tokenSaleInstance.tokensold();
47     }).then(function(amount) {
48       assert.equal(amount.toNumber(), numberoftokens, 'increments the number of tokens sold');
49       return tokenInstance.balanceOf(buyer);
50     }).then(function(balance) {
51       assert.equal(balance.toNumber(), numberoftokens);
52       return tokenInstance.balanceOf(tokenSaleInstance.address);
53     }).then(function(balance) {
54       assert.equal(balance.toNumber(), tokensAvailable - numberoftokens);
55       // Try to buy tokens different from the ether value
56       return tokenSaleInstance.buyTokens(numberoftokens, { from: buyer, value: 1 });
57     }).then(assert.fail).catch(function(error) {
58       assert(error.message.indexOf('revert') >= 0, 'msg.value must equal number of tokens in wei');
59       return tokenSaleInstance.buyTokens(80000, { from: buyer, value: numberoftokens * tokenPrice });
60     }).then(assert.fail).catch(function(error) {
61       assert(error.message.indexOf('revert') >= 0, 'cannot purchase more tokens than available');
62     });

```

Figure 51:Write token buying function

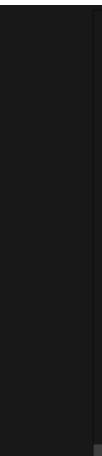


```

11
12   event Sell(address _buyer, uint256 _amount);
13
14   constructor(TrushangToken _tokenContract,uint256 _tokenPrice) {
15     //Assign an admin
16     admin = msg.sender;
17     //Token contract
18     tokenContract = _tokenContract;
19     //Token Price
20     tokenPrice = _tokenPrice;
21   }
22
23   //Multiply
24   function multiply(uint x, uint y) internal pure returns (uint z){
25     require(y == 0 || (z = x * y) / y == x);
26   }
27
28   //Buy Tokens
29   function buyTokens(uint256 _numberoftokens) public payable{
30     //Require that value is equal to tokens
31     require(msg.value == multiply(_numberoftokens,tokenPrice));
32
33     //Require that the contract has enough tokens
34     require(tokenContract.balanceOf(address(this)) >= _numberoftokens);
35
36     //keep that tokens transfer is successful
37     require(tokenContract.transfer(msg.sender, _numberoftokens));
38
39     //Keep track of tokenSold
40     tokensSold += _numberoftokens;
41     //Trigger sell event
42     emit Sell(msg.sender, _numberoftokens);
43   }
44 }

```

Figure 52:Write buytoken in contract



```

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale\truffle test
Using network 'development'.

Compiling your contracts...
> Compiling .\contracts\TrushangToken.sol
> Compiling .\contracts\TrushangTokensale.sol
> Compiling .\contracts\TrushangTokensale.sol
> Artifacts written to C:\Users\HP\AppData\Local\Temp\test--19428-x8yKGUGnXie
> Compiled successfully using:
  - solc: 0.8.0+commit.c7fd78e.Emscripten clang

Contract: TrushangToken
  ✓ initializes the contract with the correct values (68ms)
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (252ms)
  ✓ approves tokens for delegated transfer (85ms)
  ✓ handles delegated token transfers (298ms)

Contract: TrushangTokensale
  ✓ initializes the contract with the correct values
  ✓ facilitates token buying (387ms)

7 passing (1s)

C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale\]

```

Figure 53:Deploy the transaction

MNEMONIC	HD PATH
tray jungle behind they super client flight shine gospel effort uphold faith	m/44'/60'0'@account_index
ADDRESS	
0xbAaEc63DA2a292378AE8e911dceA015D6C9bea5D	BALANCE 99.93 ETH
ADDRESS	
0x48F45714598A04D5C522AFE261B48B5bDe949eB9	BALANCE 99.89 ETH
ADDRESS	
0x3f3764BE59b2A67B007D6c1d25a86279CDC697ED	BALANCE 100.00 ETH
ADDRESS	
0x6e0EdaA365E8D3B472574eE0D2111E409BBC347B	BALANCE 100.00 ETH
ADDRESS	
0x4E6D4172E251750ab0B84cd392659b3bc909FF34	BALANCE 100.00 ETH
ADDRESS	
0x5c60CFBAB80a6f26AB4CE09817570657C151661D	BALANCE 100.00 ETH
ADDRESS	
0x5378fe680BDDabe89E4Cec548cF2D2892b0f236	BALANCE 100.00 ETH

Figure 54: Ganache transaction

```

it('ends token sale', function(){
    return TrushangToken.deployed().then(function(instance) {
        //Grab token instance first
        tokenInstance = instance;
        return TrushangTokenSale.deployed();
    }).then(function(instance) {
        //Then grab token sale instance;
        tokenSaleInstance = instance;
        //Try to end sale from account other than the admin
        return tokenSaleInstance.endsale({from:buyer});
    }).then(assert.fail).catch(function(error){
        assert(error.message.indexOf('revert')>=0, 'must be admin to end sale'));
        // End Sale as admin
        return tokenSaleInstance.endsale({from:admin});
    }).then(function(receipt){
        return tokenInstance.balanceOf(admin);
    }).then(function(balance){
        assert.equal(balance.toNumber(),999990,'returns all unsold TrushangToken to admin');
        // Check that the contract is destroyed
        return web3.eth.getCode(tokenSaleInstance.address);
    }).then(function(code){
        assert.equal(code,'0x','token sale contract was destroyed');
    });
});
});

```

Figure 55: Ending Sale token function

```

// Ending Token Sale
function endsale() public {
    //Require admin
    require(msg.sender == admin);
    //Transfer remaining tokens to admin
    uint256 unsoldTokens = tokenContract.balanceOf(address(this));
    require(tokenContract.transfer(admin, unsoldTokens));

    //Destroy contract
    selfdestruct(payable(admin));
}

```

Figure 56: Ending Sale token in contract

```
C:\Users\HP\Desktop\SEM-6\BT\Practical-3\token_sale>truffle test
Using network 'development'.

Compiling your contracts...
> Compiling ./contracts/TrushangToken.sol
> Compiling ./contracts/TrushangTokenSale.sol
> Compiling ./contracts/TrushangTokenSale.sol
Artifacts written to C:\Users\HP\AppData\Local\Temp\test--17340-hxtYwxqLoF43
Compiled successfully using:
  solc: 0.8.4+commit.c7df7d8e.Emscripten.clang

Contract: TrushangToken
  ✓ initializes the contract with the correct values
  ✓ allocates the initial supply upon deployment
  ✓ transfers token ownership (226ms)
  ✓ approves tokens for delegated transfer (61ms)
  ✓ handles delegated token transfers (181ms)

Contract: TrushangTokenSale
  ✓ initializes the contract with the correct values
  ✓ facilitates token buying (157ms)
  ✓ ends token sale (102ms)

8 passing (898ms)
```

Figure 57: Deployed contract after end token sale

MNEMONIC	tray jungle behind they super client flight shine gospel effort uphold faith	HD PATH	m/44'/60'/0'/account_index
ADDRESS	0xbAaEc63DA2a292378AE8e911dceA015D6C9bea5D	BALANCE 99.96 ETH	TX COUNT 74 INDEX 0
ADDRESS	0x48F45714598A04D5C522AFE261B48B5bDe949eB9	BALANCE 99.81 ETH	TX COUNT 19 INDEX 1
ADDRESS	0x3f3764BE59b2A67B007D6c1d25a86279CDC697ED	BALANCE 100.00 ETH	TX COUNT 0 INDEX 2
ADDRESS	0x6e0EdaA365E8D3B472574eE0D2111E409BBC347B	BALANCE 100.00 ETH	TX COUNT 0 INDEX 3
ADDRESS	0x4E6D4172E251750ab0B84cd392659b3bc909FF34	BALANCE 100.00 ETH	TX COUNT 0 INDEX 4
ADDRESS	0x5c60CFBAB80a6f26AB4CE09817570657C151661D	BALANCE 100.00 ETH	TX COUNT 0 INDEX 5
ADDRESS	0xe5378fe680BDDabe89E4Cec548cF2D2892b0f236	BALANCE 100.00 ETH	TX COUNT 0 INDEX 6

Figure 58: Ganache transaction

```
package.json 1, U X
package.json > ...
1  {
2    "name" : "Trushang-Token-Sale",
3    "version" : "1.0.0",
4    "description" : "Trushang Token Sale ICO",
5    "main" : "truffle.js",
6    "directories": {
7      "test" : "test"
8    },
9    "author": "Trushang",
10   "license": "ISC",
11   "devDependencies": {
12     "lite-server": "^2.3.0"
13   }
14 }
```

Figure 59: Package. Json file

```

{
  "name": "trushang-Token-Sale",
  "version": "1.0.0",
  "description": "Trushang Token Sale ICO",
  "main": "truffle.js",
  "directories": {
    "test": {
      "test": "test"
    }
  },
  "author": "Trushang",
  "license": "ISC",
  "devDependencies": {
    "lite-server": "2.3.0"
  }
}

```

added 158 packages, and audited 159 packages in 9s
 9 packages are looking for funding
 run 'npm fund' for details
7 vulnerabilities (2 low, 2 moderate, 3 high)
 Some issues need review, and may require choosing a different dependency.
 Run 'npm audit' for details.

Figure 60:Install dependency

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Trushang Token Sale</title>
  <link href="css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container" style="width: 650px;">
    <div class="row">
      <div class="col-lg-12">
        <h1 class="text-center">TRUSHANG TOKEN SALE</h1>
        <br/>
      </div>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/web3.min.js"></script>
    <script src="js/truffle-contract.min.js"></script>
    <script src="js/app.js"></script>
  </div>
</body>
</html>

```

Figure 61:Write UI implementation

```

{
  "server": {
    "baseDir": "./src",
    "watchOptions": {
      "ignored": "node_modules"
    }
  }
}

```

Figure 62:Write bs.config.json

```

index.html JS app.js JS 1_deploy_contract.js
migrations > JS 1_deploy_contract.js > <unknown> > exports
1 const TrushangToken = artifacts.require('./TrushangToken.sol');
2 const TrushangTokenSale = artifacts.require('./TrushangTokenSale.sol');
3
4 module.exports = async function(deployer) {
5   try {
6     // Deploy Token with initial supply
7     await deployer.deploy(TrushangToken, 1000000);
8     const tokenInstance = await TrushangToken.deployed();
9
10    // Token price is 0.001 Ether
11    const tokenPrice = web3.utils.toBN('1000000000000000');
12
13    // Deploy TokenSale
14    await deployer.deploy(TrushangTokenSale, TrushangToken.address, tokenPrice);
15    const tokenSaleInstance = await TrushangTokenSale.deployed();
16
17    // Transfer tokens to TokenSale Contract (75% of total supply)
18    await tokenInstance.transfer(
19      tokenSaleInstance.address,
20      web3.utils.toBN('750000')
21    );
22
23    console.log('Token deployed at:', tokenInstance.address);
24    console.log('TokenSale deployed at:', tokenSaleInstance.address);
25    console.log('Tokens transferred to sale contract');
26
27  } catch (error) {
28    console.error('Deployment failed:', error);
29    throw error;
30  }
31};

```

Figure 63:Deploy contract implementation

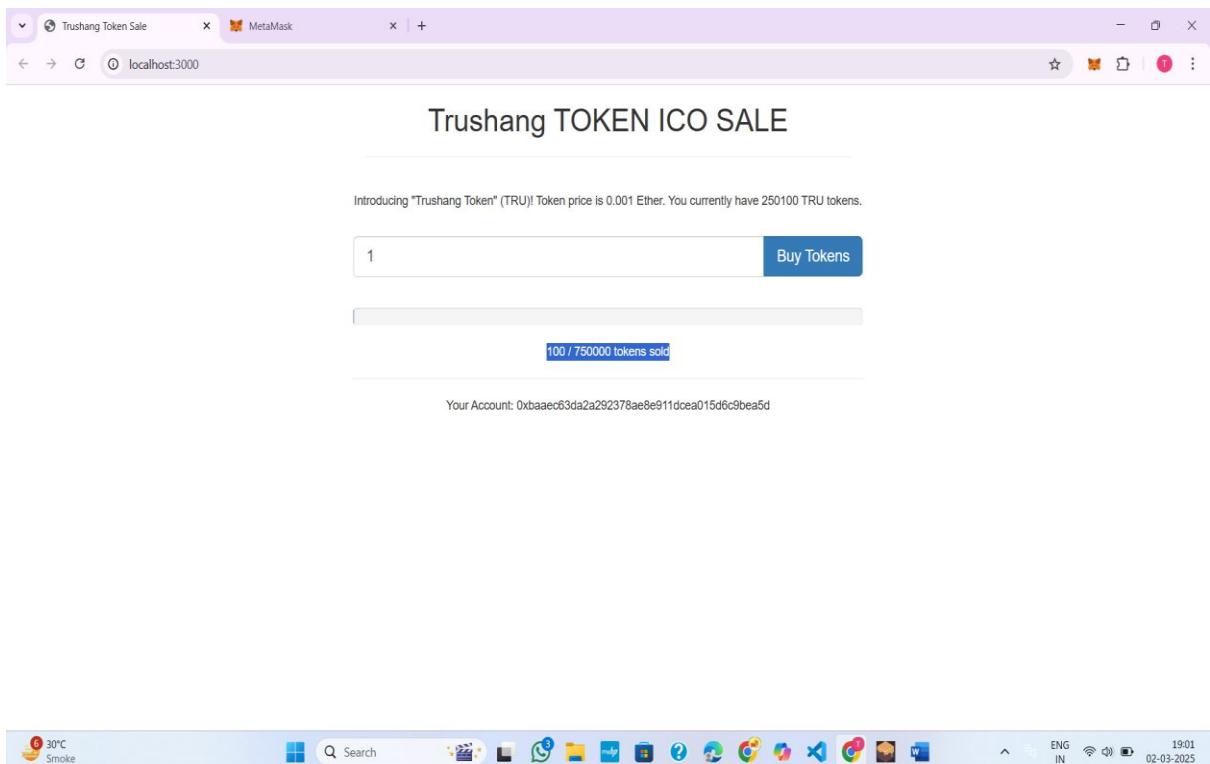


Figure 64: UI of my application

	Mar 2, 2025		
 G	Buy Tokens Confirmed	-0.1 ETH -₹18,443.24 INR	
 G	Buy Tokens Failed	-0 ETH -₹0.00 INR	
 G	Buy Tokens Failed	-0.001 ETH -₹184.43 INR	
 G	Buy Tokens Failed	-0.1 ETH -₹18,443.24 INR	
 G	Buy Tokens Failed	-0.1 ETH -₹18,443.24 INR	
 G	Buy Tokens Failed	-1 ETH -₹184,432.39 INR	
 G	Buy Tokens Failed	-1 ETH -₹184,432.39 INR	

Figure 65: Transaction history

LATEST APPLICATIONS:

- Tether USD (USDT)
- USD Coin (USDC)
- Shiba Inu (SHIB)
- Binance USD (BUSD)
- BNB (BNB)
- DAI Stablecoin (DAI)
- HEX (HEX)
- Bitfinex LEO (LEO)
- Maker (MKR)

LEARNING OUTCOME:

In this practical, I confidently designed, developed, and deployed my own ERC-20 token on the Ethereum blockchain, integrated it with front-end applications, and understood the broader implications of token economies and blockchain technology. This knowledge provided a solid foundation for further exploration of decentralized finance (DeFi), NFTs, and other blockchain-based innovations.

REFERENCES:

1. Udemy: <https://www.udemy.com/course/code-your-own-cryptocurrency/>
2. GitHub: https://github.com/Trushang-Patel/token_sale