**Basic Coin/Token Code.**

**Description:**

* This code can receive, withdraw, send all eths, withdraw specific eth(minimum one eth).
* It can also get-balance of contract as well as coins of a user.
* This can interact with only contract and the user.
* All the transactions are payable.

**Code:**

pragma solidity^0.8.15;

contract Coin

{

    address  owner;

    uint256 cionsprice= 1 ether;

    mapping (address=>uint) user;

    uint coins= 1 ether;

    constructor()

    {

        owner=msg.sender;

        user[owner]=100;

    }

    modifier ownable

    {

        require(msg.sender==owner,"You are not allowed");

        \_;

    }

    function deposit() public payable ownable

    {

        user[msg.sender]+=msg.value;

    }

    function getbalance() public view  returns  (uint)

    {

        return address(this).balance;

    }

    function BurnTokens(uint \_tokenburn) public  ownable

    {

        user[owner]-=\_tokenburn;

    }

    function transfer\_Eths(address payable reciver) public payable

    {

       reciver.transfer(msg.value);

    }

    function withdraw\_Allcontract() public payable

    {

        address payable to= payable(msg.sender);

        to.transfer(getbalance());

    }

    function withdraw\_contract() public payable

    {

        address payable to= payable(msg.sender);

        to.transfer(coins);

    }

    function purchase() public payable

    {

      require((user[owner]\*coins)/msg.value >0,"Not Enough Tokens");

       user[msg.sender]+=msg.value/cionsprice;

       user[owner]-=msg.value/cionsprice;

    }

     function Tokens\_balance() public view returns (uint)

    {

         return user[msg.sender];

    }

    function Send\_tokens(address reciever , uint \_tokens) public

    {

        require(user[msg.sender] >= \_tokens, "Not enough tokens");

        assert(user[reciever] + \_tokens >= user[reciever]);

        assert(user[msg.sender] - \_tokens <= user[msg.sender]);

        user[reciever]+=\_tokens;

        user[msg.sender]-=\_tokens;

    }

}