pragma solidity >=0.4.22 <0.6.0;

interface token {

function transfer(address receiver, uint amount) external;

}

contract Crowdsale {

address public beneficiary;

uint public fundingGoal;

uint public amountRaised;

uint public deadline;

uint public price;

token public tokenReward;

mapping(address => uint256) public balanceOf;

bool fundingGoalReached = false;

bool crowdsaleClosed = false;

event GoalReached(address recipient, uint totalAmountRaised);

event FundTransfer(address backer, uint amount, bool isContribution);

/\*\*

\* Constructor

\*

\* Setup the owner

\*/

constructor(

address ifSuccessfulSendTo,

uint fundingGoalInEthers,

uint durationInMinutes,

uint etherCostOfEachToken,

address addressOfTokenUsedAsReward

) public {

beneficiary = ifSuccessfulSendTo;

fundingGoal = fundingGoalInEthers \* 1 ether;

deadline = now + durationInMinutes \* 1 minutes;

price = etherCostOfEachToken \* 1 ether;

tokenReward = token(addressOfTokenUsedAsReward);

}

/\*\*

\* Fallback function

\*

\* The function without name is the default function that is called whenever anyone sends funds to a contract

\*/

function () payable external {

require(!crowdsaleClosed);

uint amount = msg.value;

balanceOf[msg.sender] += amount;

amountRaised += amount;

tokenReward.transfer(msg.sender, (amount / price)\*10\*\*18);

emit FundTransfer(msg.sender, amount, true);

}

modifier afterDeadline() { if (now >= deadline) \_; }

/\*\*

\* Check if goal was reached

\*

\* Checks if the goal or time limit has been reached and ends the campaign

\*/

function checkGoalReached() public afterDeadline {

if (amountRaised >= fundingGoal){

fundingGoalReached = true;

emit GoalReached(beneficiary, amountRaised);

}

crowdsaleClosed = true;

}

/\*\*

\* Withdraw the funds

\*

\* Checks to see if goal or time limit has been reached, and if so, and the funding goal was reached,

\* sends the entire amount to the beneficiary. If goal was not reached, each contributor can withdraw

\* the amount they contributed.

\*/

function safeWithdrawal() public afterDeadline {

if (!fundingGoalReached) {

uint amount = balanceOf[msg.sender];

balanceOf[msg.sender] = 0;

if (amount > 0) {

if (msg.sender.send(amount)) {

emit FundTransfer(msg.sender, amount, false);

} else {

balanceOf[msg.sender] = amount;

}

}

}

if (fundingGoalReached && beneficiary == msg.sender) {

if (msg.sender.send(amountRaised)) {

emit FundTransfer(beneficiary, amountRaised, false);

} else {

//If we fail to send the funds to beneficiary, unlock funders balance

fundingGoalReached = false;

}

}

}

}