Address:

0xc5cE95ec9D9178d62ED2F68c51bc03aBe05A5F2c

Smart Contract:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract CreditAssetAutoReset {
    address public lender;
    address public borrower;
    uint256 public yieldThreshold;
    uint256 public totalFunded;
    uint256 public constant RELEASE_AMOUNT = 0.001 ether;

    event Funded(address indexed lender, uint256 amount);
    event FundsReleased(address indexed borrower, uint256 amount);
    event BorrowerRegistered(address indexed borrower);

    constructor(uint256 _yieldThreshold) payable {
        yieldThreshold = _yieldThreshold;
        lender = msg.sender;
        if (msg.value > 0) {
            totalFunded = msg.value;
            emit Funded(msg.sender, msg.value);
        }
    }

    modifier onlyBorrower() {
        require(msg.sender == borrower, "Not the registered borrower.");
        _;
    }

    function fundLoan() external payable {
        require(msg.value > 0, "Must send ETH to fund the loan");
        totalFunded += msg.value;
        lender = msg.sender;
        emit Funded(msg.sender, msg.value);
    }

    function registerAsBorrower() public {
        require(borrower == address(0), "A borrower is already active.");
        borrower = msg.sender;
        emit BorrowerRegistered(msg.sender);
    }

    function releaseFunds(uint256 actualYield) public onlyBorrower {
        require(actualYield >= yieldThreshold, "Yield below required threshold.");
        require(totalFunded >= RELEASE_AMOUNT, "Insufficient contract balance.");
        require(address(this).balance >= RELEASE_AMOUNT, "Contract balance too low.");
        totalFunded -= RELEASE_AMOUNT;
        payable(borrower).transfer(RELEASE_AMOUNT);
        emit FundsReleased(borrower, RELEASE_AMOUNT);
        borrower = address(0);
    }

    function getStatus() public view returns (string memory) {
        if (borrower == address(0)) {
            return "Awaiting borrower registration";
        } else {
            return "Borrower registered, awaiting yield trigger";
        }
    }

    function getBalance() public view returns (uint256) {
        return address(this).balance;
    }

    function getRemainingFunds() public view returns (uint256) {
        return totalFunded;
    }

    receive() external payable {
        totalFunded += msg.value;
        lender = msg.sender;
        emit Funded(msg.sender, msg.value);
    }
}
```

ABI:

```
[
	{
		"inputs": [
			{
				"internalType": "uint256",
				"name": "_yieldThreshold",
				"type": "uint256"
			}
		],
		"stateMutability": "payable",
		"type": "constructor"
	},
	{
		"anonymous": false,
		"inputs": [
			{
				"indexed": true,
				"internalType": "address",
				"name": "borrower",
				"type": "address"
			}
		],
		"name": "BorrowerRegistered",
		"type": "event"
	},
	{
		"anonymous": false,
		"inputs": [
			{
				"indexed": true,
				"internalType": "address",
				"name": "lender",
				"type": "address"
			},
			{
				"indexed": false,
				"internalType": "uint256",
				"name": "amount",
				"type": "uint256"
			}
		],
		"name": "Funded",
		"type": "event"
	},
	{
		"anonymous": false,
		"inputs": [
			{
				"indexed": true,
				"internalType": "address",
				"name": "borrower",
				"type": "address"
			},
			{
				"indexed": false,
				"internalType": "uint256",
				"name": "amount",
				"type": "uint256"
			}
		],
		"name": "FundsReleased",
        "type": "event"
    },
    {
        "inputs": [],
        "name": "RELEASE_AMOUNT",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "borrower",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "fundLoan",
        "outputs": [],
        "stateMutability": "payable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "getBalance",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
{
        "inputs": [],
        "name": "getRemainingFunds",
        "outputs": [
                {
                        "internalType": "uint256",
                        "name": "",
                        "type": "uint256"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "getStatus",
        "outputs": [
                {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "lender",
        "outputs": [
                {
                        "internalType": "address",
                        "name": "",
                        "type": "address"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "registerAsBorrower",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
            {
                "internalType": "uint256",
                "name": "actualYield",
                "type": "uint256"
            }
        ],
        "name": "releaseFunds",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "totalFunded",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "yieldThreshold",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "stateMutability": "payable",
        "type": "receive"
    }
]
```

BYTECODE:

```
6080604052604051610780380380610780833981016040819052602091607c565b60028190555f8054600
1600160a01b03191633179055341560775734600381905560405190815233907f5af8184bef8e4b45eb9f6
ed7734d04da38ced226495548f46e0c8ff8d7d9a52490602001604051809103 90a25b506092565b5f60208
284031215608b575f5ffd5b5051919050565b6106e18061009f5f395ff3fe6080604052600436106100 9d57
5f3560e01c80637ed08cbe116100625780637ed08cbe146101b4578063833687eb146101c85780638db57
994146101dc578063ad044f49146101e4578063bcead63e146101f9578063e816c32c14610217575f5ffd5b
806312065fe01461010557806332ae6ceb1461012657806334d68282f1461013b5780634e69d5601461015
c5780637df1f1b91461017d575f5ffd5b36610101573460035f8282546100b391906106 08565b909155505
05f80546001600160a01b0319163390811790915560405134815 27f5af8184bef8e4b45eb9f6ed7734d04d
a38ced226495548f46e0c8ff8d7d9a524906020016040518091039 0a2005b5f5ffd5b3480156101 10575f5ff
d5b50475b6040519081526020015b60405180910390f35b34801561013 1575f5ffd5b5061011360025481 5
65b348015610146575f5ffd5b5061015a610155366004610621565b610231565b005b348015610167575f5
ffd5b506101706104455 65b60405161011d9190610638565b348015610188575f5ffd5b5060015461019c9
06001600160a01b031681565b60405160016001600160a01b03909116815260200161011d565b3480156101bf
575f5ffd5b5061015a6104ae565b3480156101d3575f5ffd5b506003546101135 65b61015a610546565b348
0156101ef575f5ffd5b50610113600354815 65b348015610204575f5ffd5b505f5461019c906001600160a0
1b031681565b348015610222575f5ffd5b50610113 66038d7ea4c6800081565b6001546001600160a01b0
316331461029057604051624 61bcd60e51b8152602060048201526 01c60248201527f4e6f742074686520
7265676973746572656420626f72726f7765722e00000000604482015260640 15b60405180910390fd5b6
002548110156102e25760405162461bcd60e51b815260206004820152601f602482015 27f5969656c6420
62656c6f772072657175697265642074687265736 86f6c642e006044820152606401610287565b66038d7
ea4c680006003541015610 33b5760405162461bcd60e51b815260206004820152601e60248201527f496e
737566666963696 56e7420636f6e74726163742062616c616e63652e0000604482015260640161028756 5
b66038d7ea4c6800047101561039257604051624 61bcd60e51b8152602060048201526019602482015 27f
436f6e7472616374206261 6c616e636520746f6f206c6f772e00000000000000604482015260640161028 7
565b66038d7ea4c6800060035f8282546103aa919061066d565b90915550506001546040516001600160a
01b03909116905f9066038d7ea4c680009082818181858883f19350505050158015610 3e8573d5f5f3e3d5
ffd5b5060015460405166038d7ea4c68000815260016001 60a01b03909116907f221c08a06b07a64803b37
87861a3f276212fcccb51c2e6234077a9b8cb13047a90602001604051809103 90a25060018054600160 01
60a01b0319169055565b6001546060906001600160a01b0316610490575060408051808201909152601e
81527f41776169746 96e6720626f72726f7765722072656769737472617469 6f6e000060208201529056 5b
6040518060600160405280602b81526020016106 81602b913990509056 5b60015460016001 60a01b0316
1561050757604051624 61bcd60e51b815260206004820152601d60248201527f4120626f72726f7765722
0697320616c7265616479206163746976652e000000604482015260640161028756 5b6001805460016 00
160a01b031916339081179091556040517f57c3b463895a0da2fe39978165ab8b71864d9225a91dd3bda7
184006e52cdd0c905f90a2565b5f341161059557604051624 61bcd60e51b815260206004820152601e602
48201527f4d7573742073656e642045544820746f2066756e6420746865206c6f616e00006044820152 60
6401610287565b3460035f82825461 05a6919061060856 5b909155505 05f80546001600160a01b0319163
3908117909155604051348152 7f5af8184bef8e4b45eb9f6ed7734d04da38ced226495548f46e0c8ff8d7d9
a524906020016040518091039 0a2565b634e487b7160e01b5f526011600452 60245ffd5b808201808211 1
561061b5761061b6105f4565b9291505056 5b5f60208284031215610631575f5ffd5b5035919050565b602
081525f825180602084015280602085016040850 15e5f604082850101526040601f19601f830116840101 9
15050929150505 65b818103818111156106 1b5761061b6105f456fe426f72726f7765722072656769737 46
57265642c20617761697469 6e67207969656c6420747269676765722 a2646970667358221220bc9f476576
1438f03571f7e9658aa0a3df3c6a08415c44aa38d32b7c02e8d41564736f6c634300081e0033
```