

《区块链技术》实验报告

实验名称	基于众筹合约的 Dapp 系统			指导老师	程宏兵
成员 1 姓名	温家伟	成员 1 学号	202103151422	成员 1 班级	21 级大数据分析 01
成员 2 姓名	刘易非	成员 2 学号	202103150210	成员 2 班级	21 级网络工程 01
<div>一、团队分工介绍</div> <div>二、项目概述和架构设计</div> <div>三、实现细节</div> <div>四、结果与讨论</div>					
<div>一、团队分工介绍</div> <div>1. 刘易非：合约设计师</div> <p>确定智能合约的整体架构和设计方向，负责项目的设计规划和技术指导。</p> <p>具体任务：</p> <p>完成需求分析和功能规格说明，明确智能合约的具体功能和交互逻辑。</p> <p>设计智能合约的数据结构，包括定义合约中的状态变量、事件和函数接口。</p> <p>确定智能合约的逻辑流程，与开发者协商设计细节，解决设计方面的问题并提供支持。</p> <div>2. 温家伟：开发者</div> <p>负责根据合约设计师的设计方案和文档，编写智能合约的具体代码实现。</p> <p>具体任务：</p> <p>根据合约设计师提供的设计文档，使用 Solidity 语言编写智能合约的代码。实现智能合约的各个功能模块，并确保代码的质量和可读性。用 streamlit 库和 web3 库构建前端交互，并且用 Ganache 和 Metamask 把链部署在本地。进行单元测试和集成测试，修复代码中的 bug 并优化性能。主动沟通交流，及时反馈问题并修改代码以满足设计需求。</p> <div>二、项目概述和架构设计</div> <div>1. 项目概述：</div> <p>该智能合约是一个众筹合约（CrowdFunding.sol），用于处理众筹活动的资金募集和管理。合约中包括项目的基本信息（例如标题、发起人、结束时间、目标金额、详细信息等），记录众筹的资金流向（当前已募集金额、剩余目标金额、捐赠人数等），以及记录具体资金用途（每个具体用途的内容、所需资金、赞成人数、反对人数等）。</p>					

2. 架构设计:

本项目实现了众筹合约的基本逻辑,并借助 streamlit 和 web3 库做了和前端的交互,用 Ganache 和 Metamask 把链部署在本地。以下是合约具体内容的设计。

(1) 基本信息和状态变量:

合约包含的基本信息有标题 (title)、发起人 (initiator)、结束时间 (endTime)、目标金额 (goal)、详细信息 (info) 等。

合约中维护了当前已筹集金额 (current)、剩余目标金额 (remainder)、捐赠人数 (funderNum)、众筹是否成功的状态 (isSuccessful) 等状态变量。

通过结构体 (struct) Use 来记录具体的资金用途,包括内容、所需资金、赞成人数、反对人数以及投票情况。

(2) 构造函数:

在构造函数中初始化项目的基本信息,包括标题、发起人、结束时间、目标金额、详细信息等,并初始化当前已筹集金额和剩余目标金额。

(3) 捐款 (contribute) 函数:

用户可以调用该函数向项目捐款,传入捐款金额和捐款人地址,要求捐款金额大于 0 且小于等于目标金额与已募集金额之差,捐款时间在结束时间之前,如果众筹已成功则无法再进行捐款。

更新当前已筹集金额、记录捐款人的捐款金额,并维护捐款人的信息(包括捐款金额、捐款人列表等)。

并且,在众筹成功的时候,会自动执行空投函数的逻辑,即向所有参与者发放奖励。

(4) 退款 (returnMoney) 函数:

捐款人可以调用该函数申请退款,要求捐款人存在且捐款金额大于 0,当前已筹集金额小于目标金额。

将捐款金额返还给捐款人,更新当前已筹集金额,清除捐款人的信息。

(5) 创建使用请求 (newUse) 函数:

用于创建新的使用案例(Use)。函数需要接收三个参数: _funder (资助方的地址)、_content (案例内容)、_money (资金数额)。

在函数中,首先会进行一系列的 require 断言,包括检查调用者是否为发起人 (initiator)、资金是否不超过余额 (remainder)、目标是否与当前状态 (current) 相符。如果满足所有 require 条件,将扣除相应资金,然后将新的使用案例信息(包括内容、资金、赞同数和反对数)添加到 allUses 数组中。

(6) 给使用请求投票 (voteForUse) 函数:

该函数用于为特定使用案例投票。函数接收三个参数: _funder (资助方的地址)、index (案例索引)、agree (是否赞同)。

函数中也有一系列 require 断言,如检查目标是否与当前状态相符、确保投票者未对该案例进行过投票、资助方信息的存在及资金充足等。

如果条件满足,会更新该资助方对该案例的投票状态,并根据是否赞同来更新赞同数或

反对数。若赞同数或反对数达到一定数量（超过目标数的一半），则会触发相应操作，如资金的转移或退还。

三、实现细节

1. 智能合约代码

```
// SPDX-License-Identifier: GPL3.0
pragma solidity ^0.6.0;

// 一个众筹
contract CrowdFunding {
    string public title; // 众筹标题
    address payable public initiator; // 众筹发起人
    uint public goal; // 众筹目标金额
    string public info; // 众筹详情
    uint public remainder; // 众筹剩余可用金额(成功后)
    uint public current; // 众筹已筹金额
    uint public funderNum; // 参与者数量
    bool public isSuccessful; // 众筹是否成功
    address[] public funders; // 所有参与者
    mapping(address => uint) public funderMoney; // 各参与者投的钱
    mapping(address => uint) public funderId; // 各参与者序号

    // 使用请求
    struct Use {
        string content; // 请求内容
        uint money; // 请求金额
        uint agreeNum; // 同意票数
        uint disagreeNum; // 不同意票数
        mapping(address => bool) isVote; // 每个参与者是否投票
    }
    Use[] public allUses; // 所有使用请求

    constructor(string memory _title, address payable _initiator, uint _goal, string memory _info) public
    {
        title = _title;
        initiator = _initiator;
        goal = _goal;
        info = _info;
        current = 0;
        remainder = _goal;
        funderNum = 0;
    }
}
```

```
// 往一个众筹里投钱
```

```
function contribute(uint money, address _funder) payable public{  
    require(money>0 && money<=goal-current);  
    //require(isSuccessful == false);
```

```
  
    current = current + money;  
    if(funderId[_funder] != 0){  
        funderMoney[_funder] += money;  
    }  
    else{  
        funderMoney[_funder] = money;  
        funders.push(_funder);  
        funderNum++;  
        funderId[_funder] = uint(funders.length);  
    }  
}
```

```
// 退钱
```

```
function returnMoney(address _funder) public{  
    require(funderId[_funder] != 0 && funderMoney[_funder] > 0);  
    require(current < goal);
```

```
  
    address payable funderAddr = address(uint160(_funder));  
    funderAddr.transfer(funderMoney[_funder]);  
    current -= funderMoney[_funder];  
    funderId[_funder] = 0;  
    funderMoney[_funder] = 0;  
}
```

```
// 创建使用请求
```

```
function newUse(address _funder, string calldata _content, uint _money) public{  
    require(_funder == initiator);  
    require(_money <= remainder);  
    require(goal == current);
```

```
  
    remainder -= _money;  
    allUses.push(Use({  
        content: _content,  
        money: _money,  
        agreeNum: 0,  
        disagreeNum: 0  
    }));  
}
```

```
// 给使用请求投票
```

```
function voteForUse(address _funder, uint index, bool agree) payable public{  
    require(goal == current);  
    require(allUses[index].isVote[_funder] != true);  
    require(funderId[_funder] != 0 && funderMoney[_funder] > 0);  
    require(allUses[index].agreeNum < (goal + 1) / 2);  
    require(allUses[index].disagreeNum < (goal + 1) / 2);  
  
    //require(isSuccessful == true);
```

```
  
    allUses[index].isVote[_funder] = true;  
    uint leastMoney = (goal + 1) / 2;  
    if(agree==true){  
        allUses[index].agreeNum += funderMoney[_funder];  
        if(allUses[index].agreeNum >= leastMoney){  
            initiator.transfer(allUses[index].money);  
        }  
    }  
    else{  
        allUses[index].disagreeNum += funderMoney[_funder];  
        if(allUses[index].disagreeNum >= leastMoney){  
            remainder += allUses[index].money;  
        }  
    }  
}
```

```
// 空投到众筹项目的参与者
```

```
function airdrop(uint totalAirdropAmount) public {  
    if (goal==current) {  
        // 计算每个参与者的空投份额  
        uint perFunderShare = totalAirdropAmount / funderNum;  
  
        // 遍历所有参与者并进行空投  
        for (uint j = 0; j < funderNum; j++) {  
            address funder = funders[j];  
            payable(funder).transfer(perFunderShare);  
            remainder -= perFunderShare;  
        }  
    }  
}
```

```
// 获取使用请求总数
```

```
function getUseNum() public view returns(uint){
```

```
    return allUses.length;
}
```

```
// 获取使用请求内容
```

```
function getUseContent(uint index) public view returns(string memory){
    return allUses[index].content;
}
```

```
// 获取使用请求金额
```

```
function getUseMoney(uint index) public view returns(uint){
    return allUses[index].money;
}
```

```
// 获取使用请求同意数量
```

```
function getUseAgreeNum(uint index) public view returns(uint){
    return allUses[index].agreeNum;
}
```

```
// 获取使用请求不同意数量
```

```
function getUseDisagreeNum(uint index) public view returns(uint){
    return allUses[index].disagreeNum;
}
```

```
// 获取参与者是否已对某使用请求投票
```

```
function getUseVote(uint index, address funder) public view returns(bool){
    return allUses[index].isVote[funder];
}
```

```
// 获取参与者在某个众筹里投的钱
```

```
function getFunderMoney(address funder) public view returns(uint){
    return funderMoney[funder];
}
```

```
// 获取众筹地址
```

```
function getAddress() public view returns(address payable){
    return address(this);
}
```

```
// 合约余额查询
```

```
function getContractBalance() public view returns (uint) {
    return address(this).balance;
}
```

```
fallback() payable external {}
```

```

    receive () payable external {}

}

// 所有众筹
contract AllFundings {
    address payable[] fundings;
    // 创建众筹项目
    function newFunding(string calldata title, uint goal, string calldata info) public{
        CrowdFunding funding = new CrowdFunding(title, msg.sender, goal, info);
        fundings.push(funding.getAddress());
    }

    // 给一个众筹项目投钱
    function contribute(uint i) payable public {
        address payable fundingAddr = fundings[i];
        CrowdFunding(fundingAddr).contribute(uint(msg.value), msg.sender);
        fundingAddr.transfer(msg.value);
        // 众筹成功自动执行空投
        // 这里不判断众筹成功, 而是放在 airdrop 里判断
        // if(CrowdFunding(fundingAddr).isSuccessful() == true) {
            CrowdFunding(fundingAddr).airdrop(getGoal(i) / 10);
        // }
    }

    // 退钱
    function returnMoney(uint i) payable public {
        address payable fundingAddr = fundings[i];
        CrowdFunding(fundingAddr).returnMoney(msg.sender);
    }

    // 创建使用请求
    function newUse(uint i, string calldata content, uint money) public {
        address payable fundingAddr = fundings[i];
        CrowdFunding(fundingAddr).newUse(msg.sender, content, money);
    }

    // 给使用请求投票
    function voteForUse(uint i, uint useId, bool isAgree) public {
        address payable fundingAddr = fundings[i];
        CrowdFunding(fundingAddr).voteForUse(msg.sender, useId, isAgree);
    }

    // 获取众筹总数

```

```
function getTotalNum() public view returns(uint){  
    return fundings.length;  
}
```

```
// 获取某众筹地址  
function getAddress(uint i) public view returns(address payable){  
    return fundings[i];  
}
```

```
// 获取某众筹标题  
function getTitle(uint i) public view returns(string memory){  
    return CrowdFunding(fundings[i]).title();  
}
```

```
// 获取某众筹发起人  
function getInitiator(uint i) public view returns(address payable){  
    return CrowdFunding(fundings[i]).initiator();  
}
```

```
// 获取某众筹目标金额  
function getGoal(uint i) public view returns(uint){  
    return CrowdFunding(fundings[i]).goal();  
}
```

```
// 获取某众筹当前已筹金额  
function getCurrent(uint i) public view returns(uint){  
    return CrowdFunding(fundings[i]).current();  
}
```

```
// 获取某众筹剩余可用金额  
function getRemainder(uint i) public view returns(uint){  
    // return CrowdFunding(fundings[i]).remainder();  
    return CrowdFunding(fundings[i]).getContractBalance();  
}
```

```
// 获取某众筹详细信息  
function getInfo(uint i) public view returns(string memory){  
    return CrowdFunding(fundings[i]).info();  
}
```

```
// 获取某参与者在某众筹中投的钱  
function getFunderMoney(uint i, address funder) public view returns(uint){  
    return CrowdFunding(fundings[i]).getFunderMoney(funder);  
}
```



```

// 获取某众筹的使用请求总数
function getUseNum(uint i) public view returns(uint){
    return CrowdFunding(fundings[i]).getUseNum();
}

// 获取某众筹的某使用请求内容
function getUseContent(uint i, uint j) public view returns(string memory){
    return CrowdFunding(fundings[i]).getUseContent(j);
}

// 获取某众筹的某使用请求金额
function getUseMoney(uint i, uint j) public view returns(uint){
    return CrowdFunding(fundings[i]).getUseMoney(j);
}

// 获取某众筹的某使用请求同意数量
function getUseAgreeNum(uint i, uint j) public view returns(uint){
    return CrowdFunding(fundings[i]).getUseAgreeNum(j);
}

// 获取某众筹的某使用请求不同意数量
function getUseDisagreeNum(uint i, uint j) public view returns(uint){
    return CrowdFunding(fundings[i]).getUseDisagreeNum(j);
}

// 获取某众筹的某参与者是否已经针对某使用请求投票
function getUseVote(uint i, uint j, address funder) public view returns(bool){
    return CrowdFunding(fundings[i]).getUseVote(j, funder);
}
}

```

2. 前端搭建过程及代码

(1) 搭建过程:

首先, conda 创建一个新的虚拟环境, 名为 Crowding, 并且下载依赖包:

```
conda create --name Crowding
```

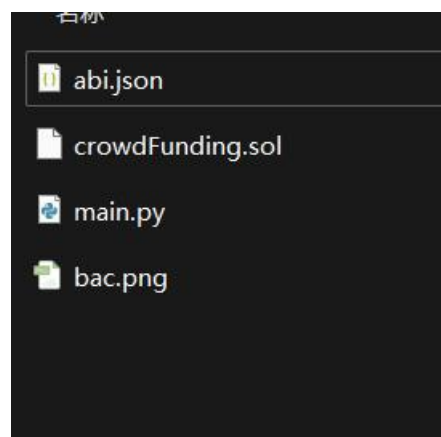
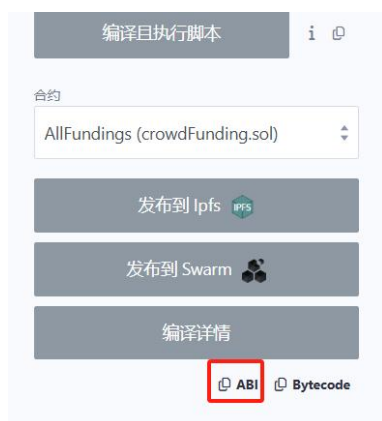
```
conda activate Crowding
```

```
pip install streamlit==1.2.0
```

```
pip install web3==4.0.0
```

```
pip install protobuf==3.19.0
```

接着, 把部署好的合约的 abi 保存到本地文件中, 命名为 abi.json



然后，根据 ganache 的钱包账户复制到 python 文件中，当然，私钥和 Metamask 的配置也要提前做好。



ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
178

GAS PRICE
2000000000

GAS LIMIT
6721975

HARDFORK
MERGE

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
IMPORTED-MONTH

SWITCH

← BACK

TX 0xd3565380abf854eca75facc0ce0efd3e6808ba9a6a0224fbeat02e3a5598dc7

SENDER ADDRESS
0x4B6D49914Ee81c3eFf7B92f02332a253F353A5F6

CREATED CONTRACT ADDRESS
0xfA0d88D6ff4E1100C5558f4D2421ABBE6a2294CA

CONTRACT CREATION

VALUE
0.00 ETH

GAS USED
3331745

GAS PRICE
197067204

GAS LIMIT
3331745

MINED IN BLOCK
168

TX DATA
0x608060405234801561001057600080fd5b50613b6b806100206000396000f3fe6080604052600436106200013a5760003560e01c8063a9cd807a11620000af578063c1c
bbca7116200006d578063c1cbbca71462000088f578063ddc2a251146200080c0578063df6ed6cb1462000913578063e76ba0111462000966578063f6b48cf014620009c357
6200013a565b8063a9cd807a1462000088f578063b7e2c93014620006f7578063b93f9b0a146200076a578063bbdb9bcd14620007d3578063bd78ea12146200083c5762000
13a565b80635486d3f11620000fd5780635486d3f146200037357806389d233ac1462000414578063929a79b1146200047157806397a24b2e146200052c578063a44da8
5b14620005ab576200013a565b8063117fb362146200013f5780631a3cd59a146200019457806326b428ec146200024f5780633785fca146200031457806337fa9209146
20003a5575b600080fd5b3480156200014c57600080fd5b5062000192600480360360608110156200016557600080fd5b8101908080359060200190929190803590602001
9092919080351515906020019092919050505062000a16565b005b348015620001a157600080fd5b50620001d160048036036020811015620001ba57600080fd5b8101908
08035906020019092919050505062000aed565b60405180806020018281038252838181518152602001915080519060200190808038360005b838110156200021357808201
5181840152602081019050620001f6565b50505050905090810190601f168015620002415780820380516001836020036101000a031916815260200191505b50925050506
0405180910390f35b3480156200025c57600080fd5b5062000296600480360360608110156200027557600080fd5b81019080803590602001909291908035906020019092
919050505062000c76565b60405180806020018281038252838181518152602001915080519060200190808038360005b83811015620002d85780820151818401526020810
19050620002b565b50505050905090810190601f168015620003065780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b
62000343600480360360208110156200032c57600080fd5b810190808035906020019092919050505062000e0b565b005b348015620003527600080fd5b506200035d620
00ece565b6040518082815260200191505060405180910390f35b3480156200038057600080fd5b5062000412600480360360608110156200039957600080fd5b81019080
803590602001909291908035906020019064010000000081115620003c157600080fd5b82018360208201115620003d457600080fd5b803590602001918460018302840
11164010000000083111715620003f757600080fd5b90919293919293908035906020019092919050505062000eda565b005b3480156200042157600080fd5b506200045b
600480360360408110156200043a57600080fd5b81019080803590602001909291908035906020019092919050505062000fdc565b6040518082815260200191505060405
180910390f35b3480156200047e57600080fd5b50620004ae600480360360608110156200049757600080fd5b8101908080359060200190929190505050620010ac565b60
405180806020018281038252838181518152602001915080519060200190808038360005b83811015620004f0578082015181840152602081019050620004d3565b5050505
0905090810190601f1680156200051e5780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b348015620005357600080fd
5b5062000593600480360360608110156200055257600080fd5b810190808035906020019092919080359060200190929190803573ffffffffffffffffffffffffffff
fffffffff1690602001909291905050506200123565b60405180821515815260200191505060405180910390f35b348015620005b57600080fd5b50620005f260048036
0040811015620005d157600080fd5b81019080803590602001909291908035906020019092919050505062001324565b604051808281526020019150506040518091039
0f35b3480156200061557600080fd5b50620006f5600480360360608110156200062e57600080fd5b8101908080359060200190640100000000811156200064c57600080
fd5b820183602082011156200065f57600080fd5b803590602001918460018302840111640100000000831117156200068257600080fd5b9091929391929390803590602
001909291908035906020019064010000000081115620006c4c57600080fd5b8035906020019184600183028401115620006e157600080fd5b9091929391929390803590602
001909291908035906020019064010000000081115620006f4c57600080fd5b80359060200191846001830284011156200070157600080fd5b9091929391929390803590602
001909291908035906020019064010000000081115620007157600080fd5b80359060200191846001830284011156200072157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200073157600080fd5b80359060200191846001830284011156200074157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200075157600080fd5b80359060200191846001830284011156200076157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200077157600080fd5b80359060200191846001830284011156200078157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200079157600080fd5b80359060200191846001830284011156200080157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200081157600080fd5b80359060200191846001830284011156200082157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200083157600080fd5b80359060200191846001830284011156200084157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200085157600080fd5b80359060200191846001830284011156200086157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200087157600080fd5b80359060200191846001830284011156200088157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200089157600080fd5b80359060200191846001830284011156200090157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200091157600080fd5b80359060200191846001830284011156200092157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200093157600080fd5b80359060200191846001830284011156200094157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200095157600080fd5b80359060200191846001830284011156200096157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200097157600080fd5b80359060200191846001830284011156200098157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200099157600080fd5b80359060200191846001830284011156200100157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200101157600080fd5b80359060200191846001830284011156200102157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200103157600080fd5b80359060200191846001830284011156200104157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200105157600080fd5b80359060200191846001830284011156200106157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200107157600080fd5b80359060200191846001830284011156200108157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200109157600080fd5b80359060200191846001830284011156200110157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200111157600080fd5b80359060200191846001830284011156200112157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200113157600080fd5b80359060200191846001830284011156200114157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200115157600080fd5b80359060200191846001830284011156200116157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200117157600080fd5b80359060200191846001830284011156200118157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200119157600080fd5b80359060200191846001830284011156200120157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200121157600080fd5b80359060200191846001830284011156200122157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200123157600080fd5b80359060200191846001830284011156200124157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200125157600080fd5b80359060200191846001830284011156200126157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200127157600080fd5b80359060200191846001830284011156200128157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200129157600080fd5b80359060200191846001830284011156200130157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200131157600080fd5b80359060200191846001830284011156200132157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200133157600080fd5b80359060200191846001830284011156200134157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200135157600080fd5b80359060200191846001830284011156200136157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200137157600080fd5b80359060200191846001830284011156200138157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200139157600080fd5b80359060200191846001830284011156200140157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200141157600080fd5b80359060200191846001830284011156200142157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200143157600080fd5b80359060200191846001830284011156200144157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200145157600080fd5b80359060200191846001830284011156200146157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200147157600080fd5b80359060200191846001830284011156200148157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200149157600080fd5b80359060200191846001830284011156200150157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200151157600080fd5b80359060200191846001830284011156200152157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200153157600080fd5b80359060200191846001830284011156200154157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200155157600080fd5b80359060200191846001830284011156200156157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200157157600080fd5b80359060200191846001830284011156200158157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200159157600080fd5b80359060200191846001830284011156200160157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200161157600080fd5b80359060200191846001830284011156200162157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200163157600080fd5b80359060200191846001830284011156200164157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200165157600080fd5b80359060200191846001830284011156200166157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200167157600080fd5b80359060200191846001830284011156200168157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200169157600080fd5b80359060200191846001830284011156200170157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200171157600080fd5b80359060200191846001830284011156200172157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200173157600080fd5b80359060200191846001830284011156200174157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200175157600080fd5b80359060200191846001830284011156200176157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200177157600080fd5b80359060200191846001830284011156200178157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200179157600080fd5b80359060200191846001830284011156200180157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200181157600080fd5b80359060200191846001830284011156200182157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200183157600080fd5b80359060200191846001830284011156200184157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200185157600080fd5b80359060200191846001830284011156200186157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200187157600080fd5b80359060200191846001830284011156200188157600080fd5b9091929391929390803590602
0019092919080359060200190640100000000811156200189157600080fd5b803590602001918460018302840111562001901576

```

# 实例化智能合约
crowd_funding_contract = w3.eth.contract(address=crowd_funding_address, abi=crowd_funding_abi)

def main_bg(main_bg):
    main_bg_ext = "png"
    st.markdown(
        f"""
        <style>
        .stApp {{
            background: url(data:image/{main_bg_ext};base64,{base64.b64encode(open(main_bg,
"rb").read()).decode()});
            background-size: cover
        }}
        </style>
        """,
        unsafe_allow_html=True
    )

def main():
    menu = ["全部众筹", "创建众筹", "使用申请", "使用申请投票"]
    main_bg('bac.png')
    choice = st.sidebar.selectbox("菜单", menu)

    # 根据选择显示不同的页面
    if choice == "全部众筹":
        all_crowdfundings()
    elif choice == "创建众筹":
        my_creations()
    elif choice == "使用申请":
        use4apply()
    elif choice == "使用申请投票":
        vote()

# 全部众筹开始
def fetch_crowdfunding_data():
    total_fundings = crowd_funding_contract.functions.getTotalNum().call()
    crowdfunding_data = []

    for i in range(total_fundings):
        title = crowd_funding_contract.functions.getTitle(i).call()
        initiator = crowd_funding_contract.functions.getInitiator(i).call()
        goal = crowd_funding_contract.functions.getGoal(i).call()
        current = crowd_funding_contract.functions.getCurrent(i).call()
        is_success = "成功" if goal == current else "进行中"

```

```

        crowdfunding_data.append({
            '名称': title,
            '发起人': initiator[:10] + '...' if len(initiator) > 10 else initiator,
            '目标金额': goal,
            '当前已筹集金额': current,
            '状态': is_success,
        })

    return pd.DataFrame(crowdfunding_data)

def display_crowdfunding_table():
    st.title("全部众筹项目")
    df = fetch_crowdfunding_data()

    # 使用 Pandas 样式设置文本居中
    styled_df = df.style.set_properties(**{
        'text-align': 'center'
    }).set_table_styles(
        [{'selector': 'th', 'props': [('text-align', 'center')]}]
    )

    # 渲染表格
    st.write(styled_df.to_html(), unsafe_allow_html=True)

def all_crowdfundings():
    display_crowdfunding_table()

# 全部众筹结束

# 创建众筹开始

def create_new_funding(title, goal, info):
    # 不知道咋用那个环境变量，我就直接传私钥了
    w3.eth.enable_unaudited_features()

    # 发送交易
    account =
w3.eth.account.privateKeyToAccount("0x38e382617e446f99e415bd8efcf5895304a04a52d1840656b58be24e03d88979")
    tx_hash = crowd_funding_contract.functions.newFunding(title, goal, info).transact({'from':
account.address})
    print("众筹项目已创建")

def my_creations():
    # 创建新的众筹项目表单
    st.title("创建新的众筹项目")
    title = st.text_input("项目标题")

```

```

goal = st.number_input("目标金额", min_value=0)
info = st.text_area("项目信息")
if st.button("创建"):
    create_new_funding(title, goal, info)
# 创建众筹结束

def use4apply():
    st.title("众筹资金使用申请")
    # 展示用户参与投资的众筹项目
    Num = st.number_input("项目编号", key="project_number", min_value=0)
    context = st.text_area("使用请求信息")
    Money = st.number_input("金额", key="request_amount", min_value=0)

    if st.button("创建"):
        # 不知道咋用那个环境变量，我就直接传私钥了
        w3.eth.enable_unaudited_features()
        # 发送交易
        account =
w3.eth.account.privateKeyToAccount("0x38e382617e446f99e415bd8efcf5895304a04a52d1840656b58be24e03d88979")
        tx_hash = crowd_funding_contract.functions.newUse(Num, context, Money).transact({'from':
account.address})
        print("使用请求已创建")

def vote():
    st.title("众筹资金使用申请投票")
    total_fundings = crowd_funding_contract.functions.getTotalNum().call()
    crowdfunding_data = []

    for i in range(total_fundings):
        num = crowd_funding_contract.functions.getUseNum(i).call()
        for j in range(num):
            title = crowd_funding_contract.functions.getTitle(i).call()
            initiator = crowd_funding_contract.functions.getInitiator(i).call()
            goal = crowd_funding_contract.functions.getGoal(i).call()
            current = crowd_funding_contract.functions.getCurrent(i).call()
            useContext = crowd_funding_contract.functions.getUseContent(i, j).call()
            useMoney = crowd_funding_contract.functions.getUseMoney(i, j).call()
            piao = crowd_funding_contract.functions.getUseAgreeNum(i, j).call()

            print(useContext)
            is_success = "成功" if goal == current else "进行中"

```

```

crowdfunding_data.append({
    '名称': title,
    '发起人': initiator[:10] + '...' if len(initiator) > 10 else initiator,
    '目标金额': goal,
    '当前已筹集金额': current,
    '状态': is_success,
    '使用请求内容': useContext,
    '使用请求金额': useMoney,
    '使用请求票数': piao,
})

df = pd.DataFrame(crowdfunding_data)

styled_df = df.style.set_properties(**{
    'text-align': 'center'
}).set_table_styles(
    [{ 'selector': 'th', 'props': [('text-align', 'center')] }]
)

# 渲染表格
st.write(styled_df.to_html(), unsafe_allow_html=True)

if __name__ == "__main__":
    main()

```

四、结果与讨论

打开我们的 Dapp 后，首先创建新的众筹任务，如下图所示，在分别填好，项目标题，目标金额，项目信息后，点击创建按钮，即可完成新的众筹任务的创建过程。

创建新的众筹项目

项目标题

希望工程

目标金额

100

项目信息

捐助希望小学

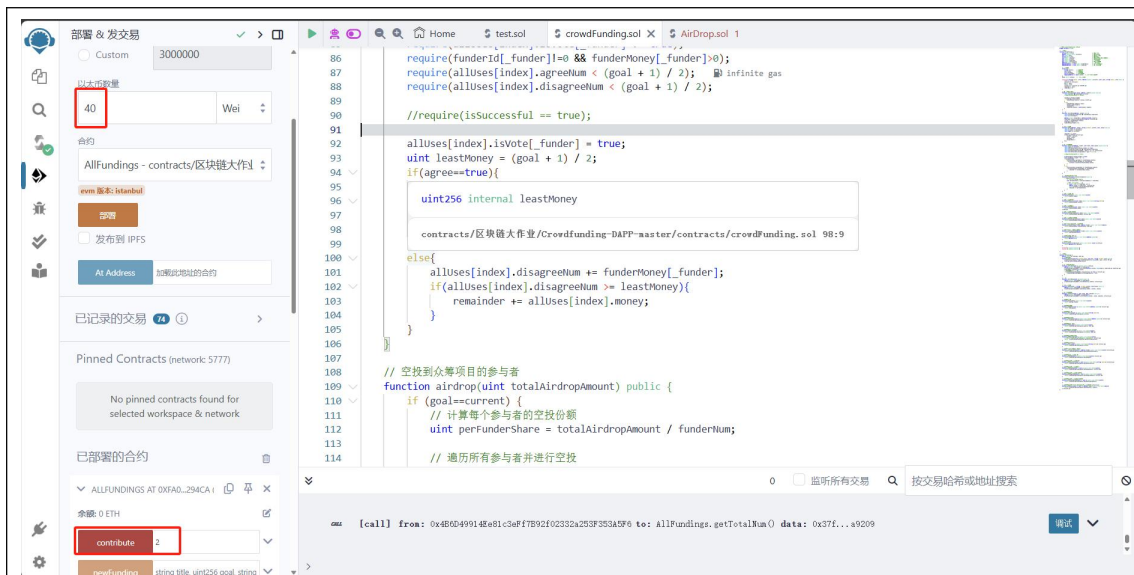
创建

与上面创建过程相同，我们继续创建了 4 个众筹项目，分别是“捐助灾区”、“救助病危老人”、“捐助残疾人”、“捐助留守儿童”。然后我们点击“全部众筹”按钮，如下图所示，此时可以看到所有的众筹信息，其中也包括了刚才发布的那四个众筹任务。

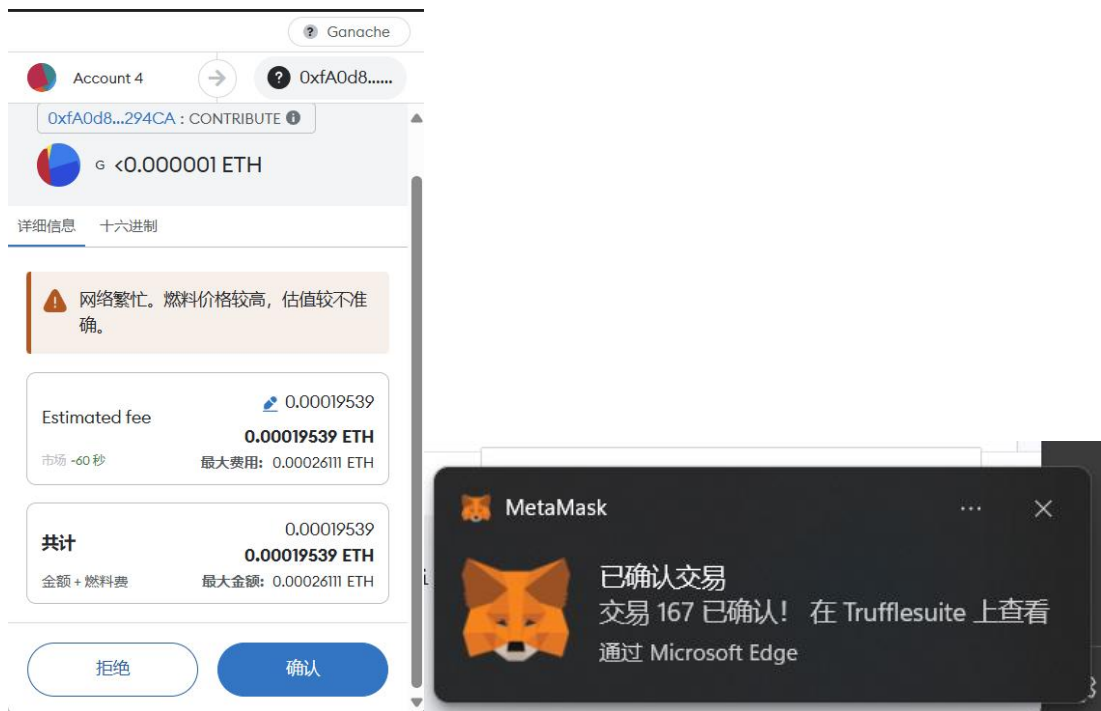
全部众筹项目

	名称	发起人	目标金额	当前已筹集金额	状态
0	希望工程	0x4B6D4991...	100	0	进行中
1	捐助灾区	0x4B6D4991...	50	0	进行中
2	救助病危老人	0x4B6D4991...	40	0	进行中
3	捐助残疾人	0x4B6D4991...	30	0	进行中
4	捐助留守儿童	0x4B6D4991...	80	0	进行中

现在这些众筹项目的状态都是进行中，我们向“救助病危老人”项目中捐一些钱，如下图所示：



点击“contribute”按钮后，MetaMask 会跳出一个界面，点击确定后，捐助成功。



这时这个众筹项目已经成功了，如下图所示。



我们在合约中写过一个空投函数，即众筹成功时会发放奖励：



我们可以看到现在合约中只剩下 36 了。

现在我们继续测试使用请求的功能，如下图所示，在填好项目编号、使用请求信息和金额后，继续点击创建按钮。



这时我们点击使用申请投票按钮，发现有使用申请。



然后我们进行投票，这里的众筹项目是一个人完成的，所以一个人投票后即可成功：

voteForUse

i: 2

used: 0

isAgree: true

Calldata 参数 transact

现在查看使用申请投票界面，票数由 0 变成了 40（有权重），投票成功！



最后测试退钱功能，我们先捐一点钱到第二个众筹项目中：



然后点击“returnMoney”按钮：



再次查看全部众筹页面时，如图所示，第二个众筹项目的当前已筹集金额从 10 变回了 0。

菜单

全部众筹

全部众筹项目

	名称	发起人	目标金额	当前已筹集金额	状态
0	希望工程	0x4B6D4991...	100	0	进行中
1	捐助灾区	0x4B6D4991...	50	0	进行中
2	救助病危老人	0x4B6D4991...	40	40	成功
3	捐助残疾人	0x4B6D4991...	30	0	进行中
4	捐助留守儿童	0x4B6D4991...	80	0	进行中

Made with Streamlit

然后我们查看 Ganache，发现所有行为均在链上有保存：

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
178

GAS PRICE
20000000000

GAS LIMIT
6721975

HARDFORK
MERGE

NETWORK ID
5777

RPC SERVER
HTTP://127.0.0.1:7545

MINING STATUS
AUTOMINING

WORKSPACE
IMPORTED-MONTH

SWITCH

BLOCK 178	MINED ON 2024-06-12 05:35:32	GAS USED 43201	1 TRANSACTION
BLOCK 177	MINED ON 2024-06-12 05:34:08	GAS USED 174350	1 TRANSACTION
BLOCK 176	MINED ON 2024-06-12 05:30:10	GAS USED 100233	1 TRANSACTION
BLOCK 175	MINED ON 2024-06-12 05:25:38	GAS USED 113368	1 TRANSACTION
BLOCK 174	MINED ON 2024-06-12 05:16:54	GAS USED 174350	1 TRANSACTION
BLOCK 173	MINED ON 2024-06-12 05:11:23	GAS USED 1558644	1 TRANSACTION
BLOCK 172	MINED ON 2024-06-12 05:10:50	GAS USED 1558388	1 TRANSACTION
BLOCK 171	MINED ON 2024-06-12 05:10:36	GAS USED 1558424	1 TRANSACTION
BLOCK 170	MINED ON 2024-06-12 05:10:15	GAS USED 1558388	1 TRANSACTION
BLOCK 169	MINED ON 2024-06-12 05:03:50	GAS USED 1575564	1 TRANSACTION

实验结果评分：