

```
// CrowdFunding
```

```
pragma solidity >=0.5.0 <0.9.0;
```

```
contract CrowdFunding{
```

```
    mapping(address=>uint) public contributors;
```

```
    address public manager;
```

```
    uint public minimumContribution ;
```

```
    uint public deadline;
```

```
    uint public target;
```

```
    uint public raisedAmount;
```

```
    uint public noOfContribution;
```

```
    struct Request{
```

```
        string description;
```

```
        address payable recipient;
```

```
        uint value;
```

```
        bool completed;
```

```
        uint noOfVoters;
```

```
        mapping(address=>bool) voters;
```

```
    }
```

```
    mapping(uint=>Request) public requests;
```

```
    uint public numRequests;
```

```
    constructor(uint _target,uint _deadline){
```

```
        target=_target;
```

```
        deadline=block.timestamp+_deadline; // seconds
```

```
        minimumContribution=100 wei;
```

```
        manager=msg.sender;
```

```
    }
```

```

function sendEth() public payable{
    require(block.timestamp<deadline,"Deadline has passed");
    require(msg.value>=minimumContribution,"Minimum Contribution is not met");

    if(contributors[msg.sender]==0){
        noOfContribution++;
    }

    contributors[msg.sender]+=msg.value;
    raisedAmount+=msg.value;
}

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

function refund() public{
    require(block.timestamp>deadline && raisedAmount<target,"You are not eligible");
    require(contributors[msg.sender]>0);
    address payable user = payable(msg.sender);
    user.transfer(contributors[msg.sender]);
    contributors[msg.sender]=0;

}

modifier onlyManager(){
    require(msg.sender==manager,"only manager can call this function");
    _;
}

```

```
function createRequest(string memory _description,address payable _recipient,uint value) public  
onlyManager{
```

```
    Request storage newRequest= requests[numRequests++];
```

```
    newRequest.description=_description;
```

```
    newRequest.recipient=_recipient;
```

```
    newRequest.value=value;
```

```
    newRequest.completed=false;
```

```
    newRequest.noOfVoters=0;
```

```
}
```

```
function voteRequest(uint _requestNo) public{
```

```
    require(contributors[msg.sender]>0,"You must be a contributor");
```

```
    Request storage thisRequest=requests[_requestNo];
```

```
    require(thisRequest.voters[msg.sender]==false, "You have already voted");
```

```
    thisRequest.voters[msg.sender]==true ;
```

```
    thisRequest.noOfVoters++;
```

```
}
```

```
function makePayment(uint _requestNo) public onlyManager{
```

```
    require(raisedAmount>target);
```

```
    Request storage thisRequest=requests[_requestNo];
```

```
    require(thisRequest.completed==false,"The request has been completed");
```

```
    require(thisRequest.noOfVoters>noOfContribution/2,"Majority does not supports");
```

```
    thisRequest.recipient.transfer(thisRequest.value);
```

```
    thisRequest.completed=true;
```

```
}
```

```
}
```