Pseudo Code:

1. Define the version of solidity >=0.5.0 < 0.9.0.

2. Create a contract CrowdFunding

2.1 Define contributors by mapping(address=>uint)

2.2 Declare minimumContribution, deadline, target, raisedAmount and noOfContributors.

2.3 Assign the address to the manager.

3. Create a structure Request.

3.1 Declare description, value and noOfVoters

3.2 Assign the address to the recipient.

4. Define a constructor function with “target” and “deadline” as its arguments.

4.1 Set the target.

4.2 Initialise deadline = block.timestamp+\_deadline

4.3 Initialise minimumContribution = 100 wei

4.4 Set manager = msg.sender

5. Define the sendEth() function with public accessibility.

5.1 require(block.timestamp < deadline)

else

print “Deadline has passed”.

5.2 require(msg.value >=minimumContribution)

else

print “Minimum Contribution is not met”.

5.3 if(contributors[msg.sender]==0)

5.3.1 increment the noOfContributor.

5.4 Add msg.value to the contributors[msg.sender]

5.5 Add msg.value to the raisedAmount.

6. Define the getContractBalance() function with public accessibility.

6.1 Return the balance of the contract.

7. Define the function refund() with public accessibility.

7.1 require(block.timestamp>deadline AND raisedAmount<target)

else

print “You are not eligible for refund”.

7.2 require(contributors[msg.sender]>0)

7.3 set address payable user = payable(msg.sender)

7.4 transfer value to contributors[msg.sender]

7.5 set the msg.sender to 0

8. Define a modifier onlyManager()

8.1 require(msg.sender==manager)

else

print “Only manager can calll this function”.

9. Define the function createRequests() with “description”, “recipient”, and “value” as its arguments.

9.1 set Request storage newRequest = requests[numRequests]

9.2 increment numRequests.

9.3 set newRequest.description=\_description;

9.4 set newRequest.recipient=\_recipient;

9.5 set newRequest.value=\_value;

9.6 set newRequest.completed=false;

9.7 set newRequest.noOfVoters=0;

10. Define voteRequest function with public accessibility and “requestNo” as argument.

10.1 require(contributors[msg.sender]>0)

else

print “You must be contributor”

10.2 set Request storage thisRequest=requests[\_requestNo]

10.3 require(thisRequest.voters[msg.sender]==false)

else

print “You have already voted”.

10.4 set thisRequest.voters[msg.sender] to true.

10.5 increment thisRequest.noOfVoters.

11. Define the function makePayment() with accessibility to manager only and with “requestNo” as argument.

11.1 require(raisedAmount>=target)

11.2 set Request storage thisRequest=requests[\_requestNo]

11.3 require(thisRequest.completed==false)

else

print “The request has been completed”.

11.4 require(thisRequest.noOfVoters > noOfContributors/2)

else

print “Majority does not support”.

11.5 transfer thisRequest.value to thisRequest.recipient.

11.6 set thisRequest.completed to true.