

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI - 590018



Project Report
on
“SECURE FUNDING TRAIL USING
BLOCKCHAIN”

Submitted in partial fulfillment of the requirements for the VIII Semester

Bachelor of Engineering
in
COMPUTER SCIENCE AND ENGINEERING
For the Academic Year
2021-2022

BY

B DHEERAJ REDDY

1PE17CS031

DEEPAK M

1PE16CS406

SHEIK MOHAMMED SUHAIL

1PE17CS423

UNDER THE GUIDANCE OF
EVLIN VIDYU LATHA
ASSISTANT PROFESSOR, Dept. of CSE, PESIT-BSC



Department of Computer Science and Engineering
PESIT - BANGALORE SOUTH CAMPUS
Hosur Road, Bengaluru - 560100

PESIT - BANGALORE SOUTH CAMPUS
HOSUR ROAD, BENGALURU - 560100
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project work entitled “SECURE FUNDING TRAIL USING BLOCKCHAIN” carried out by “B DHEERAJ REDDY, DEEPAK M, SHEIK MOHAMMED SUHAIL bearing USN’s 1PE17CS031, 1PE16CS406, 1PE17CS423” respectively in partial fulfillment for the award of Degree of Bachelors (Bachelors of Engineering) in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2021-2022. It is certified that all corrections/ suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Signature of the Guide
Prof. Evlin Vidyu Latha
Assistant Professor

Signature of the HOD
Dr. Annapurna D
HOD, CSE

Signature of the Principal
Dr. Subhash Kulkarni
Principal, PESIT-BSC

External Viva

Name of the Examiners

Signature with Date

- 1.
- 2.

Declaration

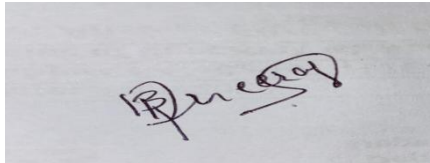
We, B DHEERAJ REDDY (1PE17CS031), DEEPAK M (1PE16CS406), SHEIK MOHAMMED SUHAIL(1PE17CS423) hereby declare that the dissertation entitles, 'SECURE FUNDING TRAIL USING BLOCKCHAIN', is an original work done by us under the guidance of Prof. Evlin Vidyu Latha, Assistant Professor, Department of Computer Science and Engineering, PESIT-BSC, Bengaluru, is being submitted in partial fulfillment for the award of B.E. in Computer Science and Engineering, VTU of the requirements for 8th semester.

Date: 26-07-2022

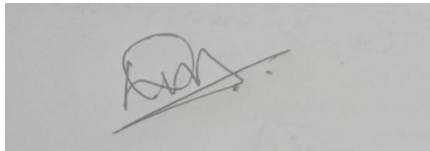
Place: Bangalore

Signature of the candidate

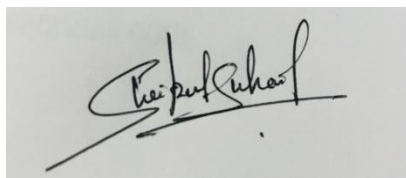
B DHEERAJ REDDY(1PE17CS031)

A handwritten signature in black ink, appearing to read 'B Dheeraj Reddy', written on a light-colored background.

DEEPAK M(1PE16CS406)

A handwritten signature in black ink, appearing to read 'Deepak M', written on a light-colored background.

SHEIK MOHAMEED SUHAIL(1PE17CS423)

A handwritten signature in black ink, appearing to read 'Sheik Mohameed Suhail', written on a light-colored background.

Acknowledgments

The satisfaction that accompany the successful completion of any task would be but incomplete without mentioning the people who made it possible, whose constant guidance crowned our effort with success.

With profound sense of gratitude, we acknowledge the guidance and the support extended by Prof Evlin Vidyu Latha, Assistant Professor, Dept. of CSE. her incessant encouragement invaluable technical support has been help in realizing this project. Their guidance gave us the environment to enhance our knowledge, skills and to reach the principal with sheer determination, dedication and hard work.

We would like to thank our principal Dr. Subhash Kulkarni for providing us the necessary facilities to carry out the project work. We express our sincere gratitude to Dr. Annapurna D, HOD, Dept. of CSE for providing us necessary facilities to carry out our project.

We are thankful to our coordinates who extended their helping hand time to time. We also extend our thanks to all faculty members of the Dept. of CSE, PESIT who have encouraged us throughout the course till now.

B DHEERAJ REDDY(1PE17CS031)
DEEPAK M(1PE16CS406)
SHEIK MOHAMMED SUHAIL
(1PE17CS423)

ABSTRACT

Crowd funding is an online money-raising strategy that began as a way for the public to donate small amount of money to help creative people finance their project. Through crowdfunding, individuals are able to invest in entrepreneurial start-ups through an intermediary, such as a broker-dealer. The problem with the current sites is they don't provide the Donor Guarantee Policy and they don't have control over the money they donated. This project is to implement crowdfunding using blockchain technology. Through this, we can provide a safe, secure and transparent way for crowdfunding.

This work of this project is to provide interactive forms for the campaign creation, donation and request approval through which both campaign creators and donors can easily create and fund the campaigns. The Donor can able to track the money that they were sent. The Blockchain will record all the transaction and store as a block.

Keywords: Crowdfunding, Blockchain, Campaign, Smart Contracts

Contents

1	Introduction	2
1.1	Definition	2
1.1.1	Blockchain	2
1.1.2	Crowdfunding	2
1.1.3	Smart Contracts	3
1.1.4	Ethereum	3
1.1.5	Solidity	4
1.1.6	Gas	4
1.1.7	Proof of work	5
1.1.8	Proof of Stake	5
1.1.9	Metamask	6
1.2	The Limits of Present-Day Crowdfunding	6
1.2.1	Platform with hectic fees	6
1.2.2	Strict rules	6
1.2.3	The requirement of marketing and advertising	6
1.2.4	Promises are broken	6
1.2.5	Vulnerability of copyright	6
1.3	How Blockchain Changes the Crowdfunding Game	7
1.3.1	Decentralization of blockchain	7
1.3.2	It is universally available	7
1.3.3	Fulfilling of promise	7
1.3.4	Variety of options	7
2	Literature Survey	8
2.1	Peer-to-Peer Lending by Alexander Bachmann[1]	8
2.2	Literature Survey From the wisdom to the wealth of crowds: A metatriangulation of crowdfunding research[2]	8
2.3	Analysis and outlook of applications of blockchain technology to equity crowdfunding in China[3]	8
2.4	Application of blockchain technology in crowdfunding by Michael	

Gebert[4]	9
2.5 Crowdsourcing and Crowdfunding Platform using Blockchain and Collective Intelligence[5]	9
3 Hardware and Software Requirements Specification	11
3.1 Hardware Requirements Specifications.....	11
3.2 Software Requirements Specifications	11
4 System Design	12
4.1 Architecture	12
4.2 Methodology.....	14
5 Implementation	16
5.1 Campaign Contract	16
5.1.1 Variables	17
5.1.2 Functions.....	17
5.1.3 Request structure	17
5.2 Compilation of contract.....	17
5.3 Deployment of contract.....	18
5.4 Backend of the project.....	19
5.5 Frontend of project.....	20
5.5.1 New Project.....	22
5.5.2 Contribute form	22
5.5.3 Request Form	22
5.5.4 Header.....	23
5.5.5 List of all projects	24
6 Result Analysis	25
6.1 Output of solidity code compilation.....	25
6.2 Output of deployment to ethereum.....	26
6.3 Frontend of project.....	26
7 Conclusion and Future Scope	30
References	30

List of Figures

4.1	Crowdfunding in a ideal world	12
4.2	Project manager creating a spending request.....	13
4.3	Voting system which ensures the money spent is in control of contributors	13
5.1	Contributor will send money to ethereum contract	16
5.2	Compilation	18
5.3	Deployment	19
5.4	Backend file.....	20
5.5	ReactJS.....	21
5.6	New project	21
5.7	Contribute form	22
5.8	Create request	23
5.9	Header	23
5.10	List of all projects	24
6.1	solidity code compilation	25
6.2	deployment to ethereum	26
6.3	Creation of project	26
6.4	List of all projects	27
6.5	Contribute to project	27
6.6	Creating a request	28
6.7	List of requests.....	28
6.8	List of requests.....	29
6.9	List of requests.....	29

Chapter 1

Introduction

1.1 Definition

1.1.1 Blockchain

A Blockchain is a growing list of blocks that are connected to one another through encryption. Square includes a cryptographic hash of the previous square, a time stamp of when it is done, and value-based information Blockchain doesn't allow any changes of the information. It is a disseminated record that is utilizes to peruse all the exchanges between the gatherings productively.

For use as a circulated record, a blockchain is ordinarily overseen by a shared system adhering to a convention for between hub correspondence and approval of squares. When the information is taken, some random square can't be changed without adjustment of resulting in blockchain, which requires understanding of the system greater part that is 51 percent.

The blockchain organize is a democratized framework. It is a mutual and un-changeable record and the data in it is open for anybody and everybody to see. Thus, anything that is based on the blockchain is straightforward and everybody included is responsible for their activities.

1.1.2 Crowdfunding

Crowdfunding is the act of financing an undertaking by collecting modest quantities of cash from countless individuals, by and large through the Internet.

This present day crowdfunding model depends on three kinds of on-screen characters: the task initiator who proposes the thought or venture to be subsidized, people

or gatherings who bolster the thought, and a stage that unites the gatherings to dispatch the idea. It is utilized to finance a wide scope of for benefit, pioneering reason, for example, innovative activities, clinical costs, travel and social business enterprise ventures.

Crowdfunding is likewise utilized for blogging and news coverage, music, free film and for subsidizing startup companies. Crowdfunding negates the standard way to deal with business fund. Generally, on the off chance that we need to raise cash-flow to begin a business or dispatch another item, we have to design our field-tested strategy, statistical surveying, and models, and afterward present thoughts around to affluent people or organizations. These subsidizing sources included banks, blessed messenger investors, venture capital firms. Crowdfunding stages gives us, the business visionary, a solitary stage to assemble, exhibit, and offer your assets, this methodology smoothes out the customary model.

1.1.3 Smart Contracts

A smart contract is a computer protocol that allows us to facilitate and verify the performance of a contract. These transactions are trackable and irreversible. Smart contracts define the rules and penalties of an agreement in the same way as a traditional contract does.

Promoters of smart contracts claim that many kinds of contractual sections may be made partially or fully self-executing and self-enforcing or self-evident. The aim of smart contracts is to provide security that is better than traditional contract and to reduce other transaction costs. Various crypto currencies is used for various types of smart contracts.

1.1.4 Ethereum

Ethereum is a decentralized framework, which implies it isn't constrained by any single administering element. A flat out lion's share of online administrations, organizations and endeavors are based on a unified arrangement of administration. This methodology has been utilized for many years, and keeping in mind that history demonstrated on numerous occasions that it's defective, its execution is as yet fundamental when the gatherings don't confide in one another.

An incorporated methodology implies single-substance control, however it additionally implies a solitary purpose of disappointment, which makes applications and online-servers using this framework incredibly helpless against programmer assaults and even force blackouts. Besides, most interpersonal organizations and other online servers expect clients to give probably some level of individual data, which is then put away on their servers. From that point, it very well may be effectively taken by the organization itself, its maverick specialists or programmers.

Ethereum, being a decentralized framework, is completely independent and isn't constrained by anybody by any means. It has no main issue of disappointment, as it is being run from a large number of volunteers' PCs around the world, which implies it can never go disconnected. Besides, clients' very own data remains on their own PCs, while content, for example, applications, recordings, and so forth., remains in full control of its makers without obeying by the guidelines forced by facilitating administrations, for example, App Store and YouTube.

1.1.5 Solidity

Solidity is an article situated, elevated level language for executing shrewd agreements. Brilliant agreements are programs which oversee the conduct of records inside the Ethereum state. Solidity was affected by C++, Python and JavaScript and is intended to focus on the Ethereum Virtual Machine (EVM).

Solidity is statically composed, bolsters legacy, libraries and complex client characterized types among other features. With Solidity you can make contracts for utilizations, for example, casting a ballot, crowdfunding, dazzle barter, and multi-signature wallets.

1.1.6 Gas

Gas refers to the fee, or pricing value, required to successfully conduct a transaction or execute a contract on the Ethereum blockchain platform. Priced in sub-units of the cryptocurrency ether, known as gwei, the gas is used to allocate resources of the ethereum virtual machine (EVM) so that decentralized applications such as smart contracts can self-execute in a secured fashion.

The exact price of the gas is determined by the network's miners, who can decline to process a transaction if the gas price does not meet their threshold. On the

ethereum blockchain, gas refers to the cost necessary to perform a transaction on the network. Miners set the price of gas and can decline to process a transaction if it does not meet their price threshold. Gas prices are denoted in gwei, which are worth 0.000000001 ether.

1.1.7 Proof of work

The way that clients identify altering by and by is through hashes, long series of numbers that fill in as verification of work. Put a given arrangement of information through a hash work (bitcoin utilizes SHA-256), and it will just ever create one hash. Due to the "torrential slide impact," in any case, even a small change to any segment of the first information will bring about an absolutely unrecognizable hash. Whatever the size of the first informational collection, the hash created by a given capacity will be a similar length. The hash is a single direction work: it can't be utilized to acquire the first information, just to watch that the information that created the hash coordinates the first information.

Creating only any hash for a lot of bitcoin exchanges would be unimportant for an advanced PC, so as to transform the procedure into "work," the bitcoin organization sets a specific degree of "trouble." This setting is balanced so another square is "mined" – added to the blockchain by producing a substantial hash – roughly at regular intervals. Setting trouble is cultivated by building up a "focus" for the hash: the lower the objective, the littler the arrangement of legitimate hashes, and the harder it is to create one.

1.1.8 Proof of Stake

Mining requires a lot of figuring capacity to run diverse cryptographic counts to open the computational difficulties. The figuring power converts into a high measure of power and force required for the proof of work. In 2015, it was evaluated that one Bitcoin exchange required the measure of power expected to control up 1.57 American family units every day. To foot the power charge, diggers would as a rule sell their granted coins for fiat cash, which would prompt a descending development in the cost of the cryptographic money.

The proof of stake (PoS) tries to address this issue by crediting mining capacity to the extent of coins held by an excavator. Along these lines, rather than using vitality

to answer PoW baffles, a PoS digger is constrained to mining a level of exchanges that is intelligent of their possession stake.

1.1.9 Metamask

Metamask is a chrome extension used for blockchain applications. Metamask is used as a key vault, secure login, and token wallet. MetaMask generates all the passwords and keys on user device, so only user can have the access to his accounts and data.

1.2 The Limits of Present-Day Crowdfunding

1.2.1 Platform with hectic fees

All the crowd funding transactions today is dependent on several different crowd funding platforms which takes lots of fees.

1.2.2 Strict rules

Many platforms serve as gatekeepers and they have strict rules and regulations.

1.2.3 The requirement of marketing and advertising

Having a great idea on a crowd funding platforms is not a guarantee that there will be a success. User will need a tactics to make their crowd funding page more visible on search engine and attract new customers to that project.

1.2.4 Promises are broken

Many of the crowd funding platforms do not ensure that the promise should be met in regards to contributors.

1.2.5 Vulnerability of copyright

Some project managers have seen their whole business collapse before they got a way to start their production. When the idea gets very popular in the crowdfunding websites, many different business people get inspired and try to make similar products like that.

1.3 How Blockchain Changes the Crowdfunding Game

1.3.1 Decentralization of blockchain

Decentralization means that no individual platforms or group of platforms control the smart contracts. It's a peer to peer network, so all the nodes in block chain have control over it.

1.3.2 It is universally available

Any one can create the project in the website with blockchain and any one who has internet connectivity can donate to the project.

1.3.3 Fulfilling of promise

Contributors do not have to worry about the empty promises like the traditional crowdfunding. The smart contracts will handle all the transactions.

1.3.4 Variety of options

Blockchain gives more freedom to project managers and the contributors. Contributors can have fractional contribution to the project.

Chapter 2

Literature Survey

2.1 Peer-to-Peer Lending by Alexander Bachmann[1]

Published by Journal of Internet Banking and Commerce in 2011

The term online shared loaning (P2P) depicts the advance beginning procedure between private people on online platforms. Initialized by bunches in online informal organizations, first business online P2P loaning stages began in 2005. Subsequently online P2P loaning is a moderately youthful exploration field.

This paper gives a short diagram of the P2P loaning business sector and surveys the examination on the determinants of P2P lending. The investigated writing gives in sights on how the determinants influence the borrowers' probability of effective financing, the last loan fee that must be paid just as the relationship of the borrowers' attributes and loaning achievement.

2.2 Literature Survey From the wisdom to the wealth of crowds: A metatriangulation of crowdfunding research[2]

Published by University College Cork in 2013 by Feller, Gleasure, and Treacy

This examination has built up a metaparadigm perspective on crowdfunding to explore the common qualities of conduct portrayed as crowdfunding, the curiosity of this wonder, and the idea of the crowdfunding research being per- formed.

2.3 Analysis and outlook of applications of blockchain technology to equity crowdfunding in China[3]

Published by journal of financial innovation in 2016

From a hypothetical viewpoint, blockchain innovation is an answer that accomplishes proficient and ease value enrollment, value exchange and move, and investor casting a ballot in the crowdfunding business, taking out legitimate dangers identified with subsidize the executives.

It can likewise assist controllers with regulating and comprehend the crowdfunding market. Therefore, blockchain applications have great possibilities in the field of value crowdfunding. At present, the blockchain application in value the executives is still in the exploratory stage; there are numerous legitimate and specialized issues to be settled.

There is as yet a requirement for blockchain undertakings and market directors to effectively help out every other, implement blockchain applications, and present imaginative arrangements.

2.4 Application of blockchain technology in crowdfunding by Michael Gebert[4]

Published by IEEE in 2017

The innovation is a progressive and problematic advancement focusing on the decrease of administration and guideline without bargaining lawful arrangements on business direct. The blockchain innovation gives a disseminated open record that upgrades straightforwardness to such an extent that members can lead issues without worries of inconvenience over the web.

In particular, blockchain innovation dispenses with data asymmetry completely along these lines fitting each partner's requirements for confirmation of legitimacy.

2.5 Crowdsourcing and Crowdfunding Platform using Blockchain and Collective Intelligence[5]

Published by International Journal of Computer Sciences and Engineering in 2019

Crowdfunding and Crowdsourcing in India are still in its earliest stages. Be that as it may, it faces a lot of challenges. Being an amazingly new idea, the Indian populace despite everything has not broadly acknowledged on the web crowdfunding or publicly supporting.

India's tremendous populace implies that India conceivably has an enormous contributor base and workforce, any semblance of which is unmatched to different nations. The essential prerequisites of any business will be business capital and human resources. This is particularly evident in the instances of startup adventures and low level organizations as these generally battle with social affair assets.

As centralization of framework gives a few advantages as for upkeep and updation of the framework furthermore, its information, it inevitably wavers with regards to the security issues. Thus utilization of blockchain in design will fortify security part of the framework.

Chapter 3

Hardware and Software Requirements Specification

3.1 Hardware Requirements Specifications

Processor - Intel CORE i5 seventh Generation

RAM - 8 GB

Processor Speed - 2.4GHz

Hard Disk - 1TB

3.2 Software Requirements Specifications

Operating System - Microsoft Windows 10

Front End - HTML , CSS , ReactJS

Back End - NodeJS

Tools - MetaMask

Language - JavaScript , Solidity

Technologies - Web3 , Ethereum Blockchain

Chapter 4

System Design

4.1 Architecture

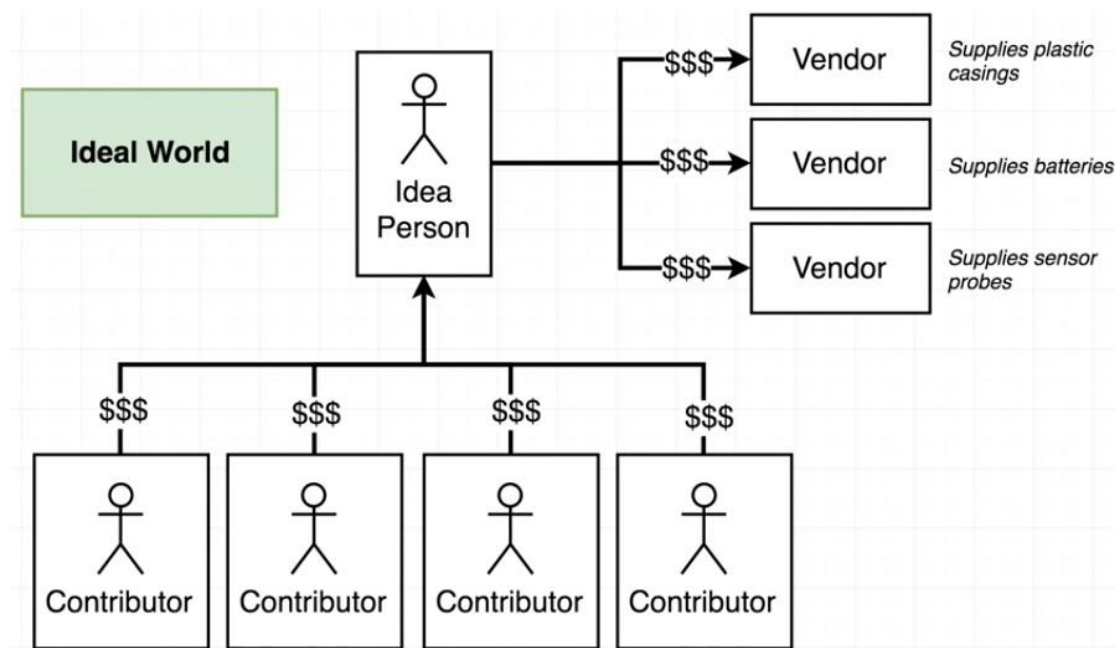


Figure 4.1: Crowdfunding in a ideal world

As shown in figure 4.1, project contains project manager, contributors, vendors, smart contract, spending request and voting system. Project manager is responsible for creating a new project by mentioning the minimum contribution to that project, the name of the project and the description of the project. After that a new project is created in the block chain with one specific address where it is stored.

Contributors then can join the project by contributing minimum contribution which project manager has set while creating the project. After contributing, contributor is

added for that project and they can accept or reject the spending request spent by the project manager.

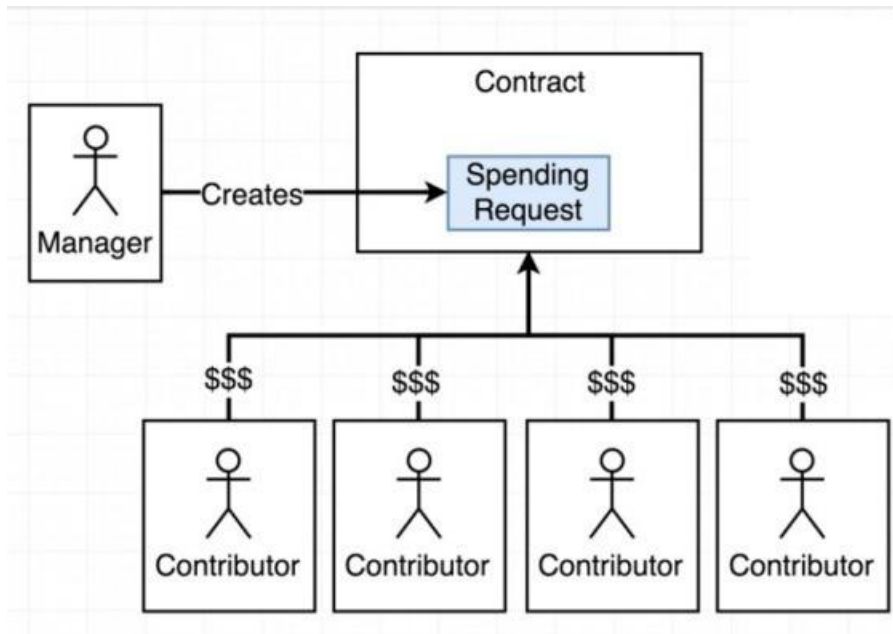


Figure 4.2: Project manager creating a spending request

If a project manager wants to spend the money contributed by contributors, then he has to create the spending request by giving the description about where he is going to spend the money, the total amount he is going to spend and the address of the vendor who will supply the things required by the project manager as shown in figure 4.2.

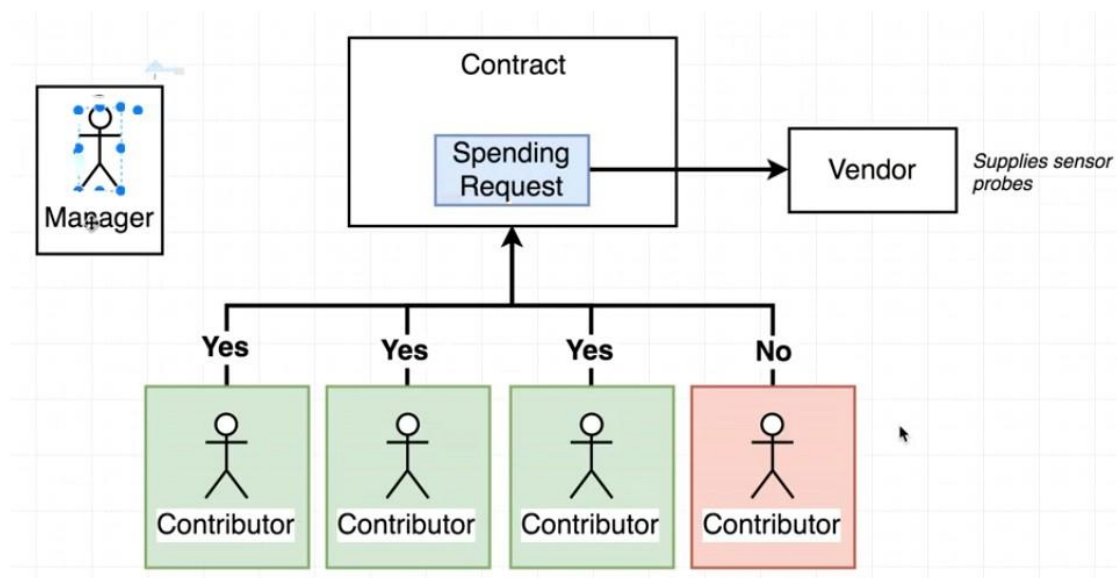


Figure 4.3: Voting system which ensures the money spent is in control of contributors

The voting system is designed so that the contributor who has contributed to that project, only he can accept or reject the spending request and the contributor once voted cannot vote again. So if more than half of the contributor agree for the spending request then the money is sent to the vendor or else it is not sent as shown in figure 4.3.

After the voting, if the majority agrees to spend the money then project manager can send that money to the vendor and the vendor will supply the utilities asked by project manager.

4.2 Methodology

- Campaign factory is a smart contract which is deployed in the ethereum blockchain when the website is first created. Ethereum returns the address where the website is stored in the blockchain.
- Campaign Factory is used to contain all the current projects which are open for contribution with their names and description about the project. These projects are created by the project managers. So this can be viewed by visiting the website and the contributor can contribute to any of these projects.
- After entering website, a user can either create their own project by clicking on create new project or user can contribute to the existing project created by others.
- If user wants to create his own project, then he can do this by clicking it on new project button on website. There they can give the project name, description and minimum contribution. After that metamask notification will pop up asking for the transaction and new project is created and the user will be project manager.
- If user wants to contribute to the existing project, then they can do this by clicking on existing project and contribute more than minimum contribution for that project. After that metamask notification will pop up asking for the transaction and the user will be contributor.
- Project manager has to create a request for spending money with the details like description about spending, money to be spent and address of the vendor. After

that metamask notification will pop up asking for the transaction and new request is created.

- The voting system is designed so that the contributor who has contributed to that project, only he can accept or reject the spending request and the contributor once voted cannot vote again.
- After this contributor can either accept or reject the request created by the project manager. If more than half of contributor accept the request then, project manager can send the money to the vendor.
- To send money to vendor, project manager has to press finalize button and then the money will be deposited to vendor's account. Metamask notification will pop up asking for the transaction to send money to vendor and then the money is sent.

Chapter 5

Implementation

5.1 Campaign Contract

A smart contract is written in the language solidity. That smart contract is fed inside the solidity compiler . The result is that compiler would be the bytecode and the interface. So in this project, a smart contract is created which contains two classes. One which stores all the instances of the project and the other which contain information about each project.

Solidity is the language which is statically compose intended for creating shrewd agreements that sudden spike in demand for the Ethereum Virtual Machine. Solidity is gathered to bytecode that is executable on the Ethereum Virtual Machine.

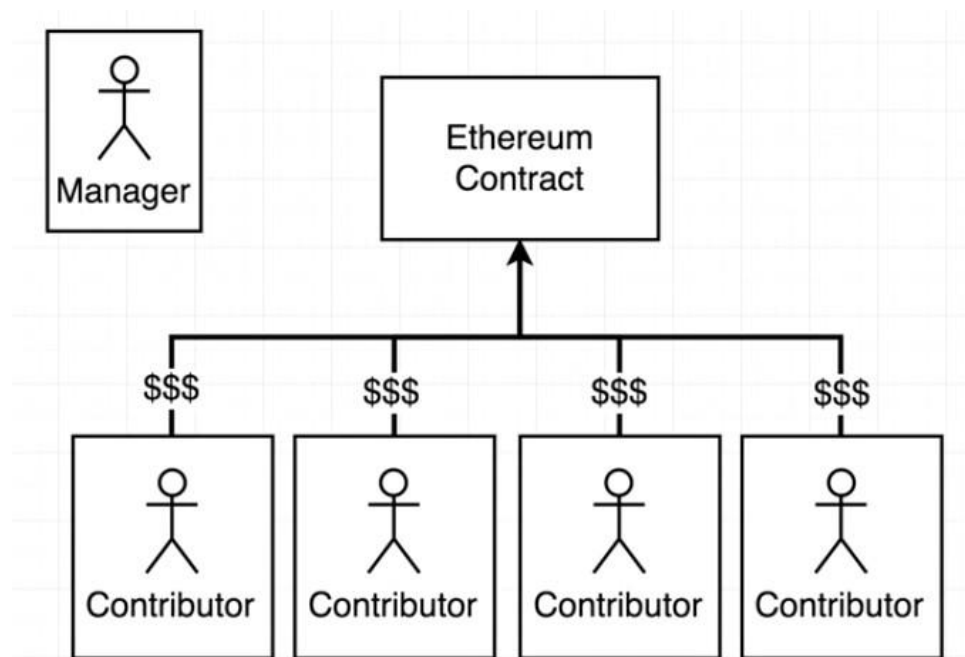


Figure 5.1: Contributor will send money to ethereum contract

In figure 5.1, it is shown that the contributor has to send money to the ethereum contract and not directly to the manager. So smart contract is created according to the requirements.

5.1.1 Variables

manager	address	address of the person who is managing the campaign
minimumContribution	unit	Minimum donation required to be considered a contributor or 'approver'
approvers	mapping	List of address for every person who has donated money
requests	Request[]	List of requests that the manager has created

5.1.2 Functions

Campaign	Constructor function that sets the minimumContribution and the owner
contribute	Called when someone wants to donate money to the campaign and become an 'approver'
createRequest	Called by the manager to create a new 'spending request'
approveRequest	Called by the contributor to approve a spending request
finalizeRequest	After a request has gotten through enough approvals, the manager can call this to get money sent to the vendor

5.1.3 Request structure

description	string	Purpose of request
amount	unit	microether to transfer
recipient	address	Who gets the money
complete	bool	Whether the request is done
approvals	mapping	Track who has voted
approvalCount	unit	Track number of approvals

5.2 Compilation of contract

In this, the Solidity code is perused utilizing the HTML way and gathered utilizing the Solc compiler. The compiler can create different yields, extending from


```

1  const path = require("path");
2  const solc = require("solc");
3  const fs = require("fs-extra");
4
5  const buildPath = path.resolve(__dirname, "build");
6  fs.removeSync(buildPath);
7
8  const campaignPath = path.resolve(__dirname, "contracts", "Campaign.sol");
9  const source = fs.readFileSync(campaignPath, "utf8");
10 const output = solc.compile(source, 1).contracts;
11
12 fs.ensureDirSync(buildPath);
13 console.log(output);
14 for (let contract in output) {
15   fs.outputJsonSync(
16     path.resolve(buildPath, contract.replace(":", "") + ".json"),
17     output[contract]
18   );
19 }
20

```

Figure 5.2: Compilation

straightforward parallels and get together over a theoretical sentence structure tree to estimations of gas utilization.

The compiled data is stored in variable called output. And each line of the compiled data is read and stored in folder named as build.

5.3 Deployment of contract

The robustness compiler transforms code into Ethereum Virtual Machine byte-code, which would then be able to be sent to the Ethereum organize as a sending exchange. Such organizations have more significant exchange charges than brilliant agreement communications and must be paid by the proprietor of the agreement.

To send a keen agreement onto the Ethereum principle net, the client should spend genuine Ether, henceforth client needs to make a record first. There are two sorts of Ethereum account. Agreement accounts hold keen agreements and are altogether constrained by code.

In order to deploy and interact contracts on a public Ethereum network such as Rinkeby via Go-Ethereum, user need an encrypted json key for the account they created through Metamask. This is the account they want to be charged for deploying and interacting with the campaign contract.

User can generate this JSON key by exporting the private key for our account in Metamask to a file and then importing it via geth.

```
const HDWalletProvider = require("truffle-hdwallet-provider");
const Web3 = require("web3");
const compiledFactory = require("../build/CampaignFactory.json");

const provider = new HDWalletProvider([
  "employ network panther love escape fiction fly skull round meadow airport bunker",
  "https://rinkeby.infura.io/v3/5c67e3609a2f4ec78482efeb82d24af1"
]);
const web3 = new Web3(provider);

const deploy = async () => {
  const accounts = await web3.eth.getAccounts();

  console.log("Attempting to deploy from account", accounts[0]);

  const result = await new web3.eth.Contract(
    JSON.parse(compiledFactory.interface)
  )
    .deploy({ data: "0x" + compiledFactory.bytecode })
    .send({ from: accounts[0] });

  console.log("Contract deployed to", result.options.address);
};
deploy();
```

Figure 5.3: Deployment

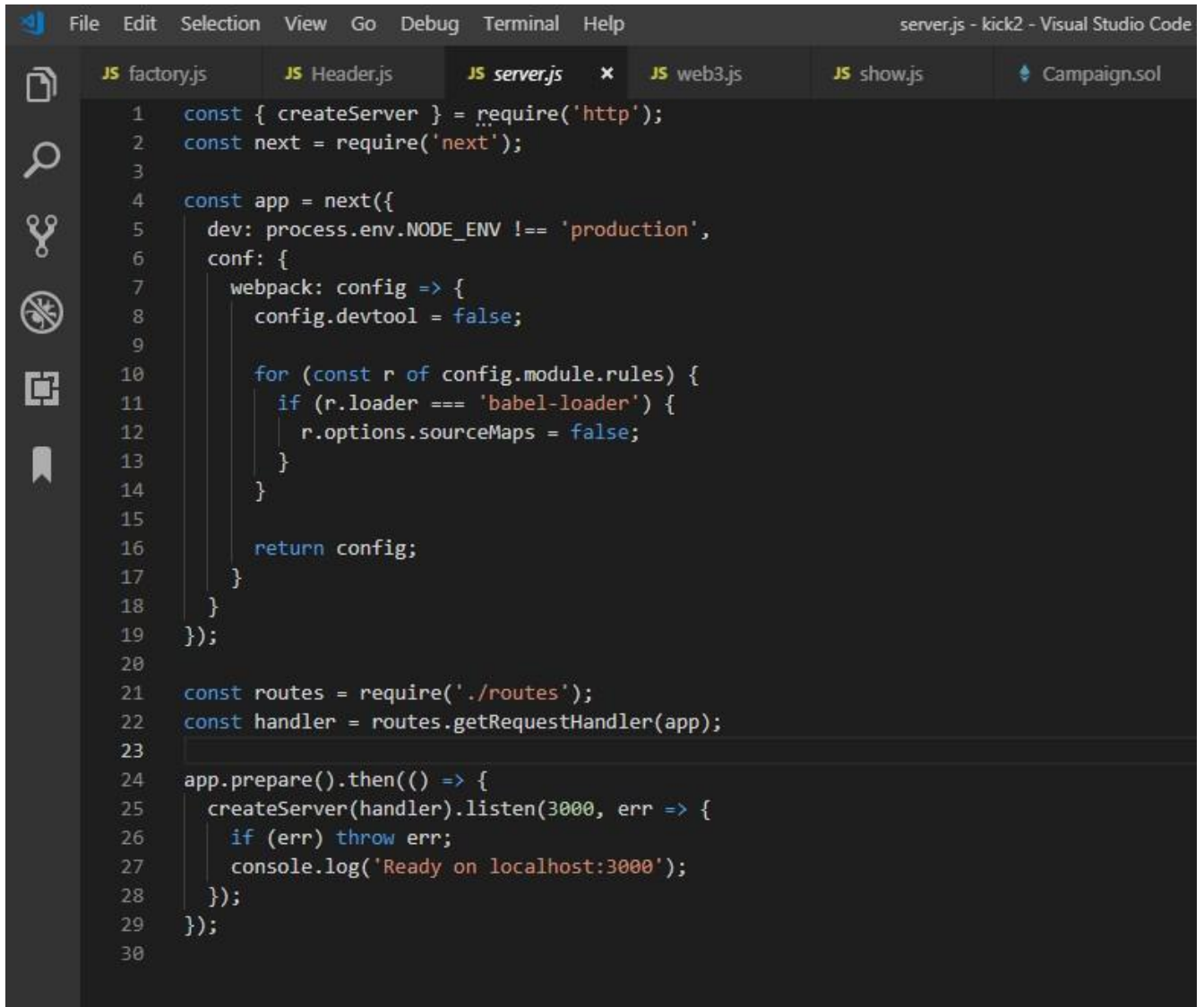
5.4 Backend of the project

Backend is the server-side of the site. It stores and organizes information, and furthermore ensures everything on the customer side of the site works fine. It is the piece of the site that you can't see and associate with. It is the bit of programming that doesn't come in direct contact with the clients.

The Next.js router allows you to do client-side route transitions between pages, similarly to a single-page application. A React component called Link is provided to do this client-side route transition.

Webpack is an open-source JavaScript module bundler. It is a module bundler essentially for JavaScript, yet it can change front-end resources like HTML, CSS,

and pictures if the relating loaders are incorporated. Webpack takes modules with conditions and produces static resources speaking to those modules.



```

1  const { createServer } = require('http');
2  const next = require('next');
3
4  const app = next({
5    dev: process.env.NODE_ENV !== 'production',
6    conf: {
7      webpack: config => {
8        config.devtool = false;
9
10       for (const r of config.module.rules) {
11         if (r.loader === 'babel-loader') {
12           r.options.sourceMaps = false;
13         }
14       }
15     },
16     return config;
17   });
18 }
19 });
20
21 const routes = require('./routes');
22 const handler = routes.getRequestHandler(app);
23
24 app.prepare().then(() => {
25   createServer(handler).listen(3000, err => {
26     if (err) throw err;
27     console.log('Ready on localhost:3000');
28   });
29 });
30

```

Figure 5.4: Backend file

5.5 Frontend of project

For the front end of the project, ReactJS and NextJS is used to retrieve the data from the bytecode.

React is a JavaScript library for building UIs. It is kept up by Facebook and a network of individual designers and organizations. React can be utilized as a base in the improvement of single-page or portable applications. Be that as it may, React is just worried about rendering information to the DOM, thus making React

applications as a rule requires the utilization of extra libraries for state the executives and steering.

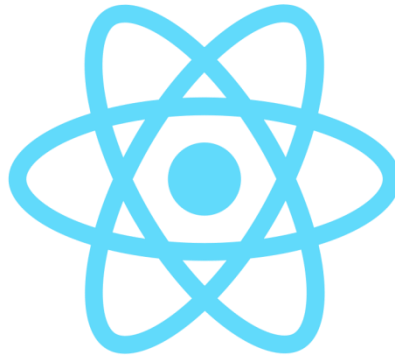


Figure 5.5: ReactJS

NextJS gives programmed code-parting, document framework based directing, hot code reloading, and general rendering. It gives unlimited authority over Babel and Webpack. It likewise has an adjustable server, directing, and next-modules. Streamlined for littler form size, quicker dev arrangement, and many different up-grades.

```

1  import React, { Component } from "react";
2  import { Form, Button, Input, Message } from "semantic-ui-react";
3  import Layout from "../components/Layout";
4  import factory from "../ethereum/factory";
5  import web3 from "../ethereum/web3";
6  import { Router } from "../routes";
7
8  class CampaignNew extends Component {
9    state = {
10      minimumContribution: "",
11      pn: "",
12      pd: "",
13      errorMessage: "",
14      loading: false
15    };
16
17    onSubmit = async event => {
18      event.preventDefault();
19
20      this.setState({ loading: true, errorMessage: "" });
21
22      try {
23        const accounts = await web3.eth.getAccounts();
24        await factory.methods
25          .createCampaign(
26            this.state.minimumContribution,
27            this.state.pn,
28            this.state.pd
29          )
30          .send({
31            from: accounts[0]
32          });
33      }

```

Figure 5.6: New project

5.5.1 New Project

Project manager is responsible for creating a new project by mentioning the minimum contribution to that project, the name of the project and the description of the project. After that a new project is created in the block chain with one specific address where it is stored as shown in fig 5.6.

```
import React, { Component } from 'react';
import { Form, Input, Message, Button } from 'semantic-ui-react';
import Campaign from '../ethereum/campaign';
import web3 from '../ethereum/web3';
import { Router } from '../routes';

class ContributeForm extends Component {
  state = {
    value: '',
    errorMessage: '',
    loading: false
  };

  onSubmit = async event => {
    event.preventDefault();

    const campaign = Campaign(this.props.address);

    this.setState({ loading: true, errorMessage: '' });

    try {
      const accounts = await web3.eth.getAccounts();
      await campaign.methods.contribute().send({
        from: accounts[0],
        value: web3.utils.toWei(this.state.value, 'microether')
      });
    }
  };
}
```

Figure 5.7: Contribute form

5.5.2 Contribute form

Contribute form contains the form to contribute to a particular project with minimum contribution. If user gives minimum contribution, user is added as a contribute to that project and renders back to home page as shown in fig 5.7.

5.5.3 Request Form

If a project manager wants to spend the money contributed by contributors, then he has to create the spending request by giving the description about where he is going to spend the money, the total amount he is going to spend and the address of the vendor who will supply the things required by the project manager.

The voting system is designed so that the contributor who has contributed to that project, only he can accept or reject the spending request and the contributor once voted cannot vote again. So if more than half of the contributor agree for the spending request then the money is sent to the vendor or else it is not sent as shown in fig 5.8.

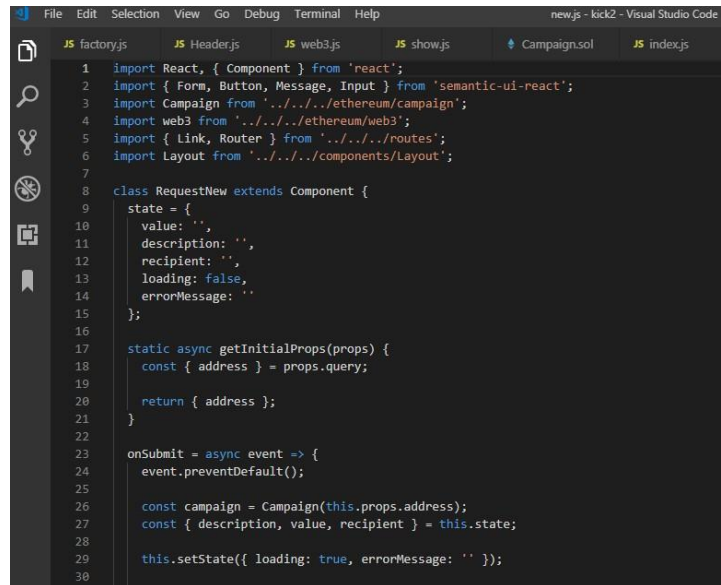


Figure 5.8: Create request

5.5.4 Header

A site header is the territory is one of the most significant zones of your site. It stumbles into the highest point of the page and shows up on each page of your site. It contains the logo and the add new project button as shown in fig 5.9.

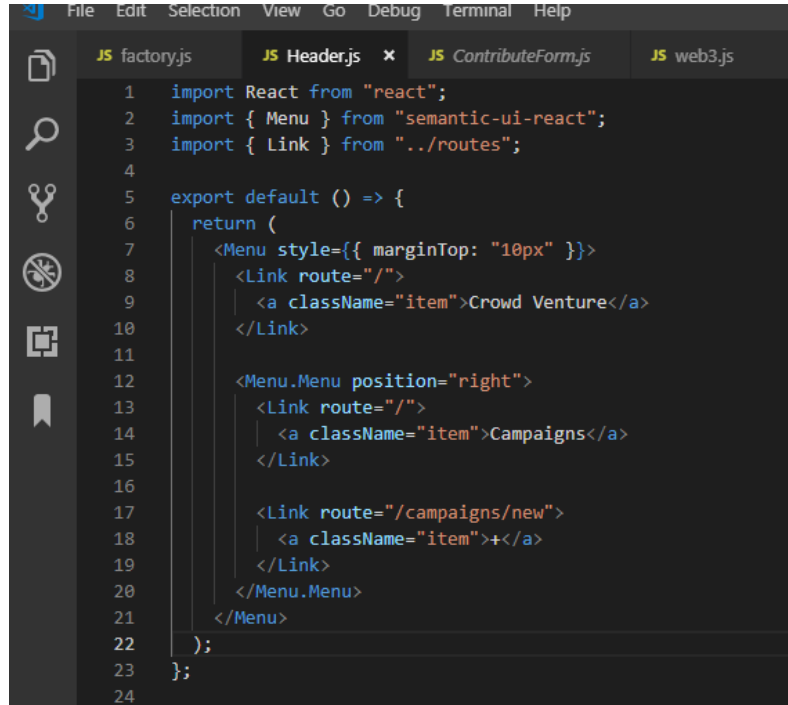


Figure 5.9: Header

5.5.5 List of all projects

Campaign factory is a smart contract which is deployed in the ethereum blockchain when the website is first created. Ethereum returns the address where the website is stored in the blockchain.

Campaign Factory is used to contain all the current projects which are open for contribution with their names and description about the project. These projects are created by the project managers. So this can be viewed by visiting the website and the contributor can contribute to any of these projects.

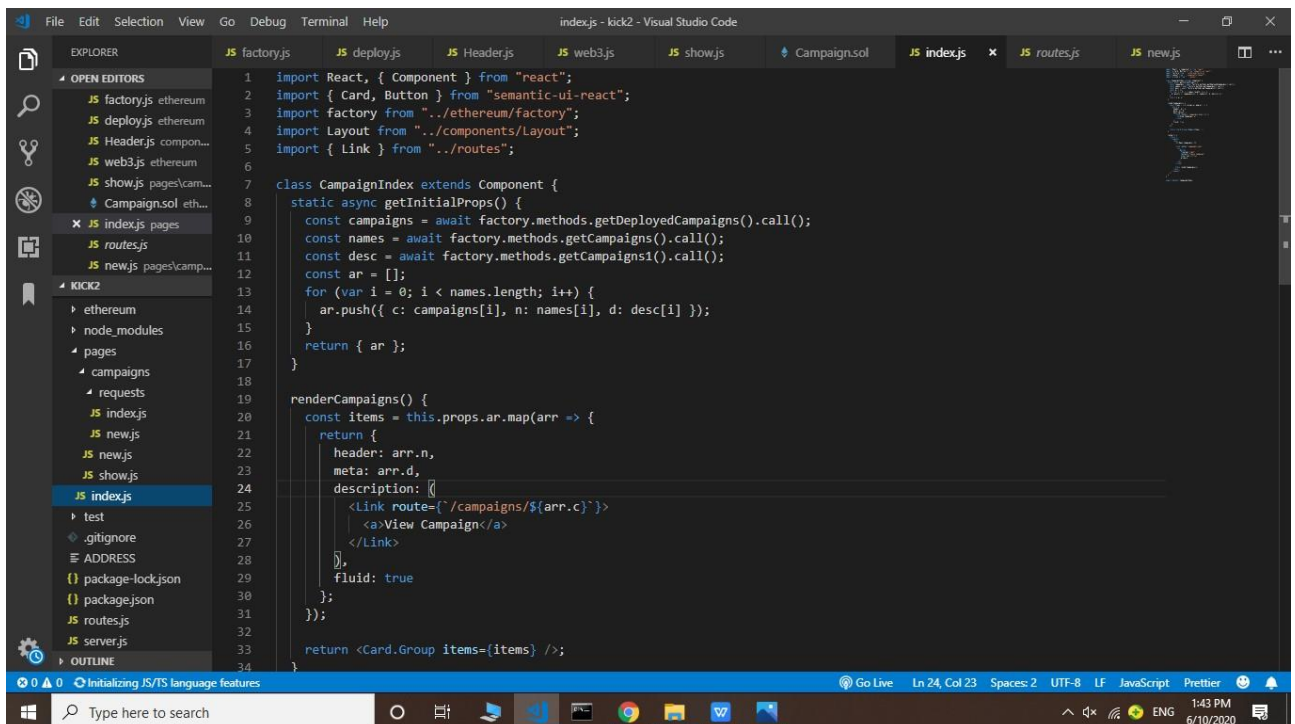
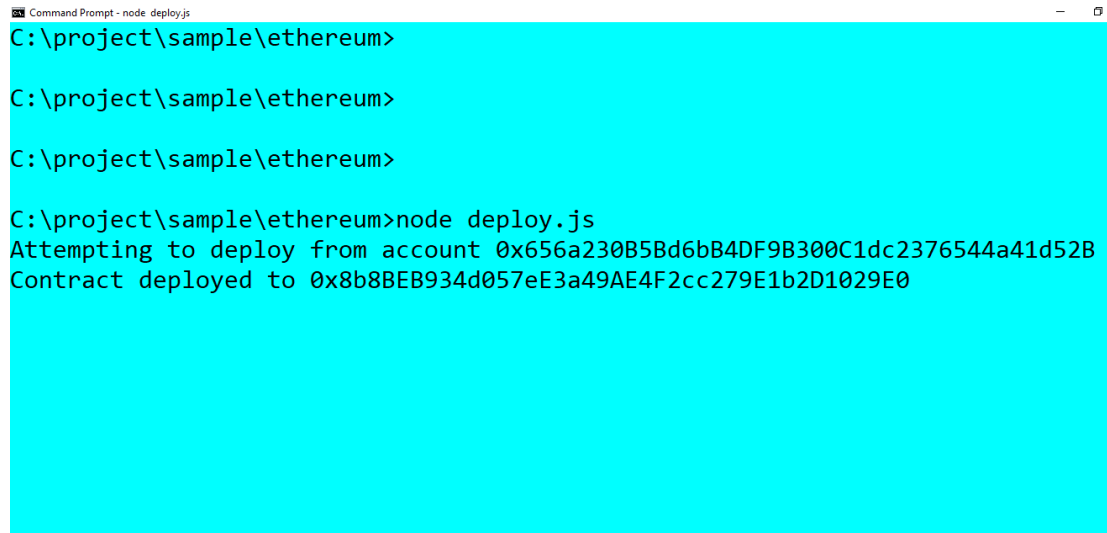


Figure 5.10: List of all projects

6.2 Output of deployment to ethereum

The contract is deployed into ethereum network using one of the address of the user. After deploying successfully, the address of where the contract is deployed in the ethereum is returned.



```
Command Prompt - node deploy.js
C:\project\sample\ethereum>
C:\project\sample\ethereum>
C:\project\sample\ethereum>
C:\project\sample\ethereum>node deploy.js
Attempting to deploy from account 0x656a230B5Bd6bB4DF9B300C1dc2376544a41d52B
Contract deployed to 0x8b8BEB934d057eE3a49AE4F2cc279E1b2D1029E0
```

Figure 6.2: deployment to ethereum

6.3 Frontend of project

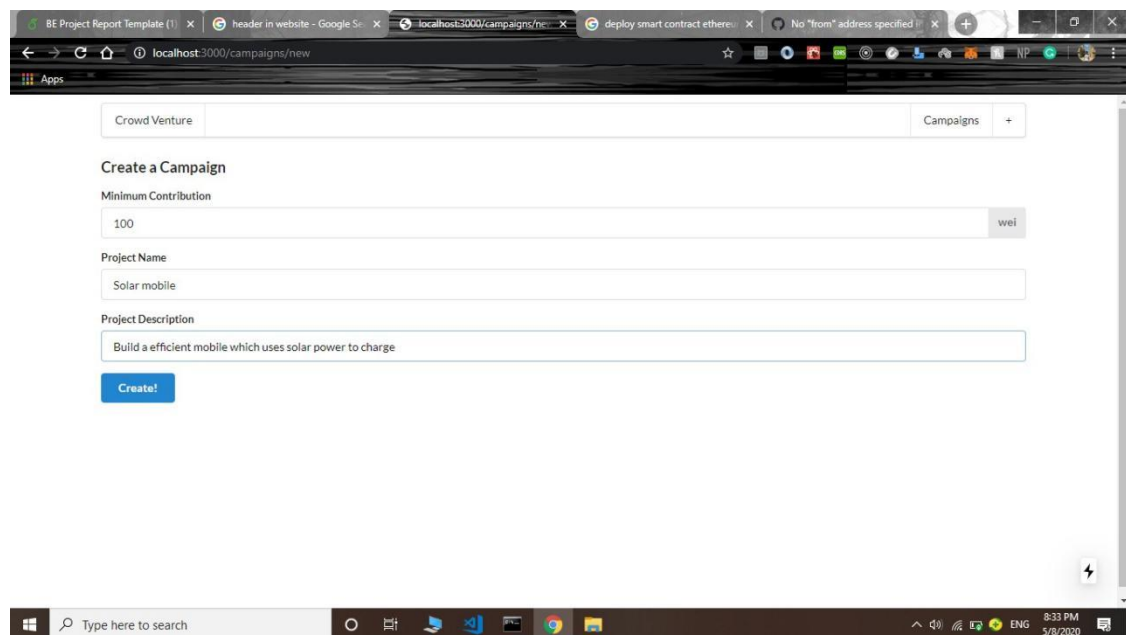


Figure 6.3: Creation of project

When a user wants to create a new project then he can do it by pressing this button. It consists of name of the project, description of the project and the minimum contribution to the project as shown in fig 6.3.

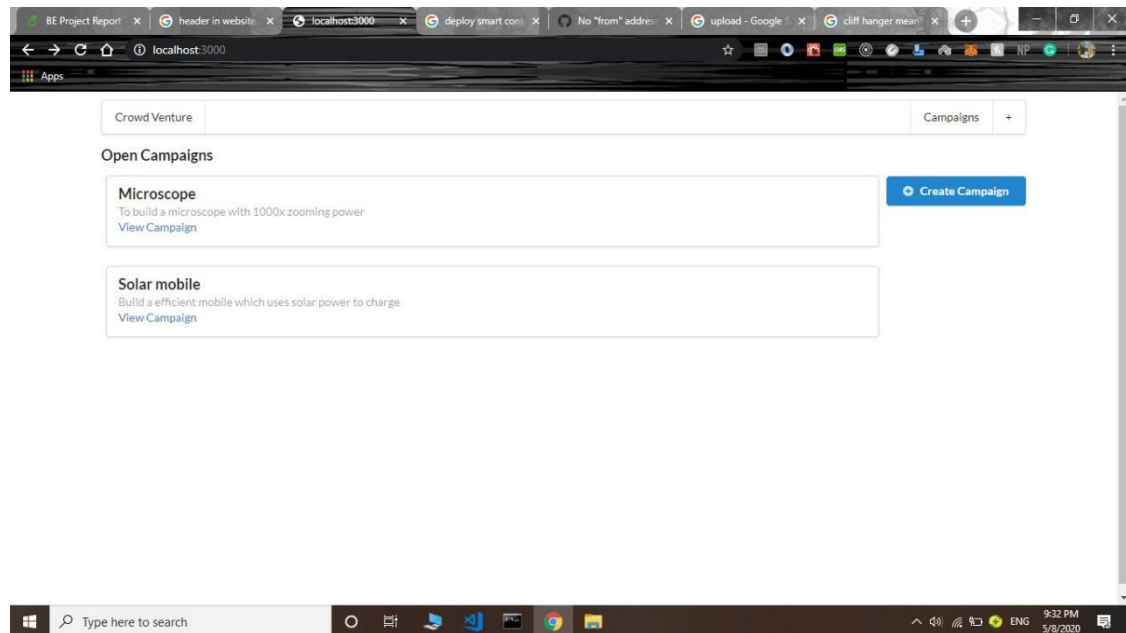


Figure 6.4: List of all projects

Here the list of all current projects are shown with their name , description and the address of project manager as shown in fig 6.4.

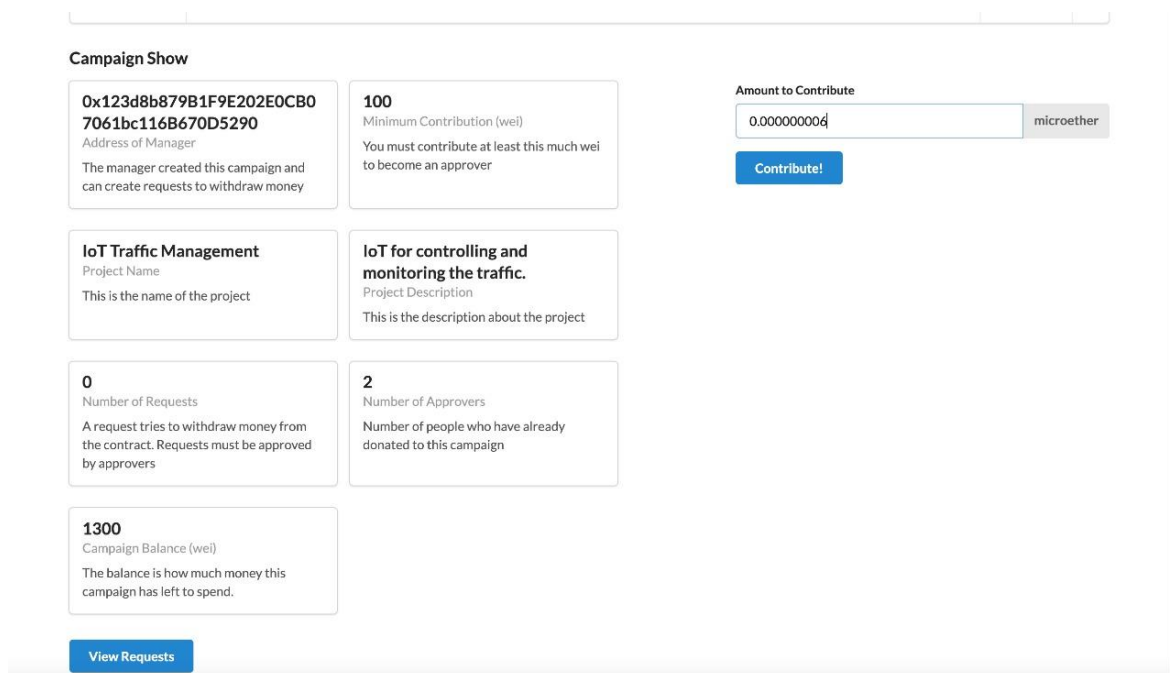


Figure 6.5: Contribute to project

Contribute form contains the form to contribute to a particular project with minimum contribution. If user gives minimum contribution, user is added as a contribute to that project and renders back to home page as shown in fig 6.5.

Crowd Venture Campaigns +

Back

Create a Request

Description
To buy mobile screen

Value in Ether
0.0001

Recipient
0x469d35eD0E1A4d842FF47AD2db2746b14036060e

Create

Figure 6.6: Creating a request

Request form is created by the project manager to request money to spend. It consists of the description of request, value in ether and address of vendor as shown in fig 6.6.

Crowd Venture Campaigns +

Requests

Add Request

ID	Description	Amount	Recipient	Approval Count	Approve	Finalize
0	To buy mobile screen	0.0001	0x469d35eD0E1A4d842FF47AD2db2746b14036060e	0/2	Approve	Finalize

Found 1 requests.

Figure 6.7: List of requests

This contains the list of requests created by the project manager with approve and finalize buttons as shown in fig 6.7.

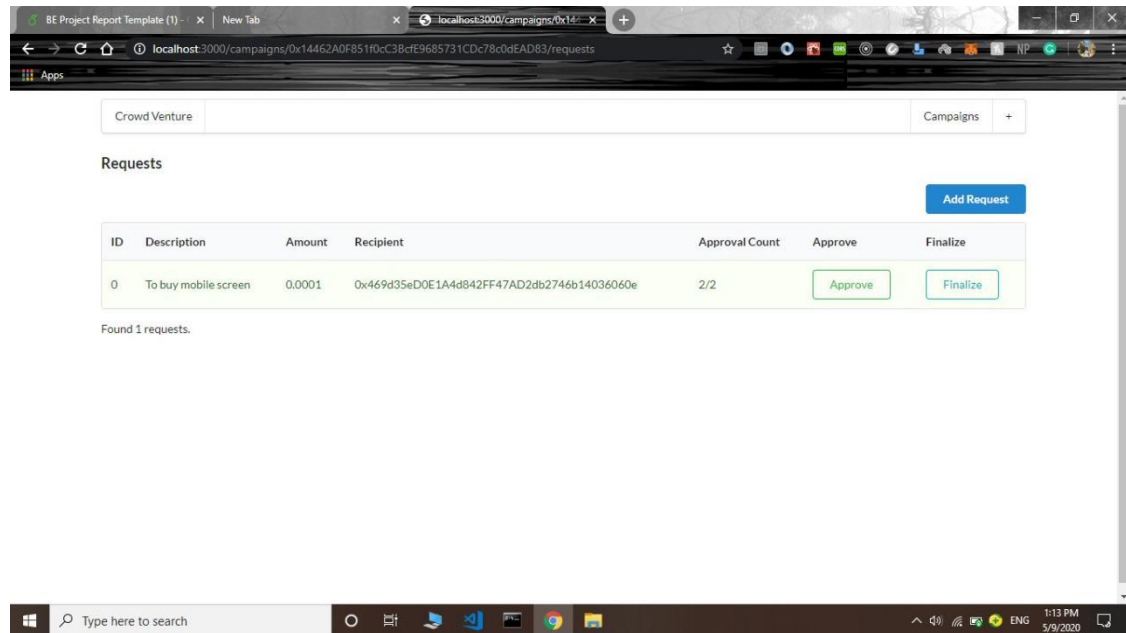


Figure 6.8: List of requests

The request turns green as soon as the number of contributors is more than half as shown in fig 6.8.

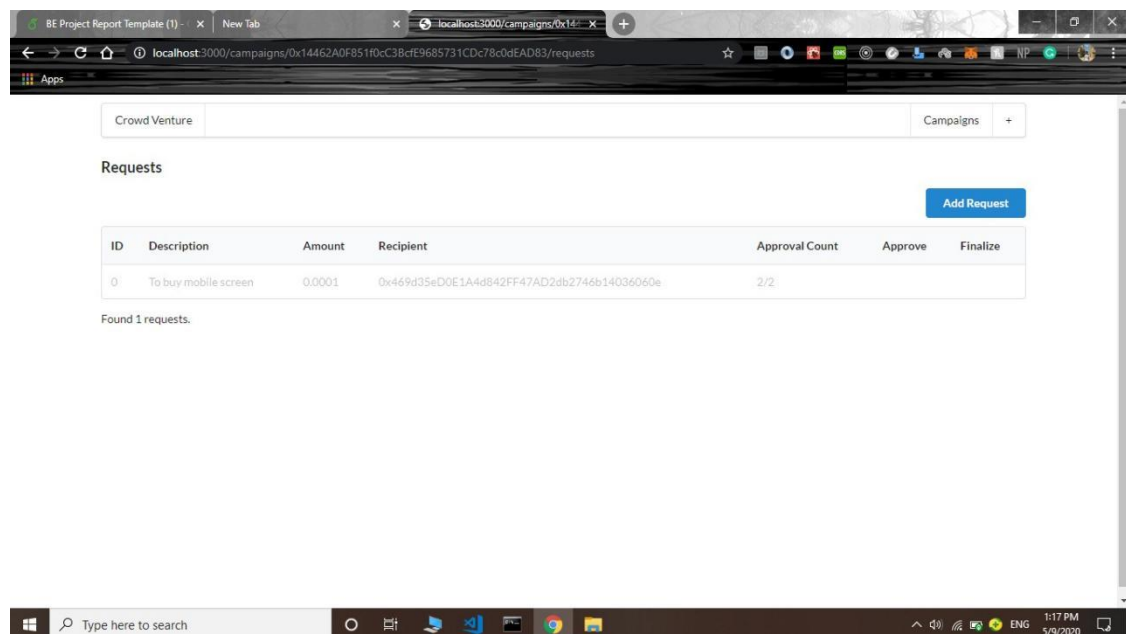


Figure 6.9: List of requests

After clicking finalize button, the money is sent to the vendor and the request is disabled as shown in fig 6.9.

Chapter 7

Conclusion and Future Scope

Blockchain in crowdfunding is a relatively new concept to the community. Since Blockchain makes the subsidizing procedure safe and offers totally straightforward access from anyplace on the planet, crowdfunding stages that utilization Blockchain can assist with augmenting the achievement of a task.

Till now, we have successfully wrote a solidity code for campaign contract and complied it using solidity compiler. The output of solidity compiler was bytecode and interface which is deployed into the ethereum blockchain using metamask. After deploying the project a decentralized web app is created with frontend for creating a new project, contributing to a project, creating a new request, approving a request and finalizing a request.

At present, the blockchain application in value the executives is still in the exploratory stage; there are numerous lawful and specialized issues to be settled. There is as yet a requirement for blockchain venture and market chiefs to effectively help out one another, execute blockchain applications, and present imaginative arrangements. They have to develop their comprehension of blockchain innovation, its worth, its chances, and its dangers.

With the development of Blockchain, our application has a promising future and lots of room for growth. We hope to make the process of bringing all ideas to life through our crowdfunding application even simpler and safer in the future.

References

- [1] *Peer-to-Peer Lending* , *Journal of Internet Banking and Commerce*(2011); Alexander Bachmann
- [2] *The wisdom to the wealth of crowds: A metatriangulation of crowdfunding research* , *University College Cork* (2013) ; Feller, Gleasure, and Treacy
- [3] *Analysis and outlook of applications of blockchain to equity crowdfunding* , *Journal of financial innovation*(2016) ; Huasheng Zhu and Zach Zhizhong Zhou
- [4] *Application of blockchain technology in crowdfunding* , *IEEE* (2017) ; MichaelGebert
- [5] *Crowdsourcing and Crowdfunding Platform using Blockchain and Collective Intelligence* , *International Journal of Computer Sciences and Engineering* (2019); Waheeda Dhokley, Saurabh Gupta, Ganesh Pawar, Abrar Shaikh

TEAM MEMBERS

1.



Name: Deepak M
USN: 1PE16CS406

Phone No: 9060014569
Email ID: deepakjain84787@gmail.com

Address: #57, Sanjeevini Nagar, Mudalapalya,
Bangalore, 560072

2.



Name: Sheik Mohammed Suhail
USN: 1PE17CS423

Phone No: 7259759971
Email ID: suhailsheik0701@gmail.com

Address: #3211 CV Road 2nd Eidgha
Mysore 570015

3.



Name: B Dheeraj Reddy
USN: 1PE17CS031

Phone No: 96118 03161
Email ID: dheeraj3836@gmail.com

Address: #250 , lake shore homes Kasavanahalli
Bangalore, 560035