**A Project Proposal**

**on**

**“DECENTRALIZED CROWDFUNDING”**



***Submitted To***

**Department of Computer and Software Engineering**

**School of Engineering**

**Pokhara University**

In the Partial Fulfillment of the

Requirements for the degree of Bachelor in Computer/Software Engineering of the Pokhara University

***Submitted By***

Dipak Sharma [19070105]

Atrip Limbu [19070096]

Rabin Rana [19070117]

Krish Gurung [19070108]

**(May 2023)**

**DECENTRALIZED CROWDFUNDING**

**Submitted By**

Dipak Sharma [19070105]

Atrip Limbu [19070096]

Rabin Rana [19070117]

Krish Gurung [19070108]

**Submitted To**

Computer and Software Engineering Department

School of Engineering

Pokhara University

# Certificate

The undersigned certify that they have read and recommended to the Department of Computer and Software Engineering for acceptance, a project report entitled “Decentralized Crowdfunding”, submitted by Name of Student(s) in partial fulfillment of the requirement for the Bachelor’s degree in computer Engineering/Software Engineering.

**Supervisor:**

...................................

<Name of Supervisor>

<Designation >

**External Examiner:**

...................................

<Name of External examiner>

<Designation>

# Acknowledgement

This project has helped us to widen our technical expertise and would have not been possible without the assistance of our mentor and guides. It has been excellent learning experience building this application. We would like to thank our project guide Dr. Uday Raj Dhungana and Bheshma Thapa. For giving us this opportunity to build this application and enhance our application development skills required for gaining a competitive edge over other candidate in this job market. Dr. Uday Raj Dhungana has been supportive and motivating to drive us towards completion of this project. All his valuable inputs has helped us to navigate through the project building phase and improve our approach and technologies. We would like to acknowledge and appreciate Mr. Bheshma Thapa for his efforts in reviewing the project report and for giving valuable feedback.

We would like to thank our friends for their valuable suggestions for this applications. Finally we would like to acknowledge all the supportive people who hepls us directly or indirectly to develop our project and helped us to keep focus intact towards achieving our goal. There are innumerable friends who helped us during this project, we are greatful to them.

Dipak Sharma [19070105]

Atrip Limbu [19070096]

Rabin Rana [19070117]

Krish Gurung [19070108]

# Abstract

Crowdfunding is a new and innovative method for funding various kinds of campaigns, wherein individual founders of the campaigns can request for funds. The campaigns may be working for profit motive, cultural or social. There are people around the globe who have small amounts of money that they can invest/help in ideas that might grow bigger or solve problems faced by them in real world, but for campaigner it’s very challenging task to reach out these donors. To solve this problem there are crowdfunding sites such as [Kickstarter](https://www.kickstarter.com/) and [Indiegogo](https://www.indiegogo.com/) which act as a platform for these campaigners to pitch their ideas to donors.

Our application replaces intermediaries such as kickstarter to provide a platform for campaigners to raise funds and for donors to donate in projects that interest them. The paper shall first deal with the role of technology in crowdfunding, followed by the various crowdfunding platforms that have emerged in recent times. Block chain is a unique, independent and a transparent technology which keeps the transaction between parties transparent. Crowdfunding is based on the trust between the donor and campaigner.

Crowdfunding platforms using the block chain technology increase the credibility of various projects and campaigns and therefore attracts huge funds from donors or investors. There are smart contract built according to the requirements. Smart contracts are deployed on block chain, they contain steps and policies that are executed according to the requirements meets.

This application is built using Remix IDE, Visual studio code editor, programming languages: JavaScript and solidity. It uses Ethereum block chain network for storing and processing smart contract transactions.

Keywords:

Crowdfunding, BlockChain Technology, Smart Contracts

# Table of Contents

[Certificate iii](#_Toc57303920)

[Acknowledgement iv](#_Toc57303921)

[Abstract v](#_Toc57303922)

[Table of Contents vi](#_Toc57303923)

[List of Tables viii](#_Toc57303924)

[List of Figures ix](#_Toc57303925)

[Abbreviations x](#_Toc57303926)

[Introduction 1](#_Toc57303927)

[1.1 Background 1](#_Toc57303928)

[1.2 Problem Statement 1](#_Toc57303929)

[1.3 Objectives 1](#_Toc57303930)

[1.4 Significance of Project 2](#_Toc57303931)

[1.5 Application 2](#_Toc57303932)

[Literature Review 3](#_Toc57303933)

[2.1 Topic as Per Your Requirement 3](#_Toc57303934)

[2.2 Topic as Per Your Requirement 3](#_Toc57303935)

[2.1.1 Subtopic as Per Your Requirement 3](#_Toc57303936)

[2.1.2 Subtopic as Per Your Requirement 3](#_Toc57303937)

[2.3 Related Task 3](#_Toc57303938)

[Methodology 4](#_Toc57303939)

[3.1 Topics as per your project……… 4](#_Toc57303941)

[3.1.1 Subtopics as required 4](#_Toc57303945)

[3.1.2 ……… 4](#_Toc57303946)

[3.2 ……… 4](#_Toc57303947)

[3.2.1 ……… 4](#_Toc57303948)

[3.2.2 ……… 4](#_Toc57303949)

[Planning 5](#_Toc57303950)

[4.1 Feasibility Analysis 5](#_Toc57303953)

[4.1.2 Operational Feasibility 5](#_Toc57303954)

[4.1.3 Technical Feasibility 5](#_Toc57303955)

[4.1.4 Economic Feasibility 5](#_Toc57303956)

[4.2 Budget Analysis 5](#_Toc57303957)

[4.3 System Requirement 5](#_Toc57303958)

[4.3.1 Hardware Requirement 5](#_Toc57303959)

[4.3.2 Hardware Requirement 5](#_Toc57303960)

[4.4 Work Schedule 6](#_Toc57303961)

[4.5 Expected Outcome 6](#_Toc57303962)

[References 7](#_Toc57303963)

# List of Tables

Table 1.1: ----------------------------------------------------------------------------------------------------------------- 2

Table 1.2: ----------------------------------------------------------------------------------------------------------------- 3

Table 2.1: ----------------------------------------------------------------------------------------------------------------- 7

# List of Figures

[Figure 3.1: ---------------------------------------------------------------------------------------------------------------- 2](#_Toc327252912)

[Figure 3.2: ---------------------------------------------------------------------------------------------------------------- 7](#_Toc327252912)

[Figure 5.1: ---------------------------------------------------------------------------------------------------------------- 9](#_Toc327252912)

<All the Table Should Be Auto generated>

# Abbreviations

|  |  |
| --- | --- |
| IPPR | Image Processing and Pattern Recognition |
| CIIL | Central Institute of Indian Languages |
| IE | Information Extraction |
| IR | Information Retrieving |
| …… | …………….. |
| …… | …………….. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Chapter 1**

# Introduction

## 1.1 Background

Crowdfunding is basically the practice of funding a project or start-up through raising money from various individuals. It is decentralized application based on Ethereum blockchain platform that allows users to donate any amount of money to the campaign/project/social-work they interested in. By using Blockchain we can provide trust, security and secure transactions and also can make sure that the donors engage in low-risk support of new campaigns and campaigns creators can gain more supports globally making it easy for them to raise large amount of funds in minimal time. Especially in blockchain world at present, there are lot of projects created by individuals or small-distributed teams that want to rise funds. Crowdfunding platform simplifies the whole idea of rising capital with help of global public that might be interested in the campaign.

## 1.2 Problem Statement

In traditional fundraising : banks and venture capital are the main way to fill the gap in funding chain. A startup founder would approach a bank or a venture capitalist with his project and ask for funding, and if anyone want to donate he/she may physically visit and deposit money to campaigner account and it bit time consuming and have many limitations.This process of fund rising requires huge amount of time, money and valuable resources that campaigner from remote location do not have access.

As of now we also have fund rising websites like kickstarts which are centralized, no trust and not fully secure. Due to above reasons fund rising on those website is also risky.

## 1.3 Objectives

Few established crowdfunding platforms such as Kickstarter and Indiegogo have revolutionized the start-up world with the flexibility and efficiency in raising funds. Blockchain based crowdfunding might be next step in evolution of fundraising platforms assisting start-up founders in the journey of their dream idea.

Blockchain have features like open source, trust less, decentralized and immutable, these features can help us to solve limitations of current crowdfunding platforms. Blockchain based crowdfunding platform can help this process by decentralizing the funding model.

Blockchain’s distributed ledger helps in getting rid of the centralized intermediaries such as Kickstarts that takes huge amounts of money from a campaign as a maintenance fee. Blockchain crowdfunding is purer form of crowdfunding as it removes any intermediaries between donors and campaigners.

Crowdfunding dapp allows campaigner to post their campaigns and then ask for funds from a community of interested people. Once the funding is successful then all the funds are directly transferred to the campaigner without any fund deduction. All the transactions are accounted and kept track of by blockchain, immutable ledger, and thus it is impossible to forge. Blockchain also gets rid of influence and manipulation done by the centralized crowdfunding platforms that have more than required access to the campaigns running in their platforms.

## 1.4 Significance of Project

Decentralized crowdfunding, also known as crowdfunding on blockchain platforms, has several advantages over traditional centralized crowdfunding models. Here are some key key points highlighting the significance of decentralized crowdfunding :

1. Elimination of intermediaries:

Decentralized crowdfunding removes the need for intermediaries, such as banks or crowdfunding platforms, which often introduce additional fees, delays and limitations. By leveraging blockchain technology, crowdfunding campaigns can directly connect campaigners and donors, enabling peer-to-peer transaction.

1. Increase transparency:

Blockchain technology provides transparency and immutability, allowing every transaction and contribution to be recorded on a public ledger. This transparency builds trust among participants, as they can verify the authenticity of the transactions and ensure that funds are being used as intended.

1. Global accessibility:

Decentralized crowdfunding platforms are accessible to anyone with an internet connection, irrespective of geographical boundaries. This global accessibility enables creators to reach a wider audience and donors to support projects from anywhere in the world, fostering inclusivity and diversity.

1. Reduced costs and fees:

Traditional crowdfunding platforms often charge substantial fees for hosting campaigns and processing transactions. In decentralized crowdfunding, costs can be significantly reduced since the platform is typically decentralized and operated by the community. Consequently, more funds can reach the project initiators, enhancing their chances of success.

1. Smart contract automation:

Blockchain-based crowdfunding platforms leverage smart contracts, which are self-executing contracts with predefined rules encoded on the blockchain. Smart contracts automate various processes, such as fund distribution, milestone-based funding, and investor protection. This automation eliminates the need for intermediaries and ensures that funds are released only when predefined conditions are met.

## 1.5 Applications

Decentralized crowdfunding, also known as decentralized fundraising, refers to the practice ofraising funds for projects or cmapaigns through blockchain based platforms without the involvement of intermediarise. Here are some applications of decentralized crowdfunding :

1. Start-up Funding:

Decentralized crowdfunding platforms enable startups and entrepreneurs to raise capital for their projects directly from a global pool of investors. It provides access to a wider range of potential funders and removes geographical barriers, increasing the chances of securing funding for innovative ideas.

1. Social Causes and Non-Profit Organizations:

Decentralized crowdfunding allows non-profit organizations and social initiatives to raise funds for charitable causes, disaster relief efforts, community development projects, and more. It enables transparent and accountable distribution of funds, ensuring donors can see how their contributions are being utilized.

1. Creative Projects:

Artists, musicians, filmmakers, and other creative individuals often face challenges in securing funding for their projects. Decentralized crowdfunding platforms offer a decentralized and democratic way for creators to engage directly with their audience, allowing them to fund their work and maintain creative control.

1. Gaming and Esports:

Decentralized crowdfunding platforms can support the development of video games, esports teams, and gaming-related projects. It allows gamers and enthusiasts to support their favorite projects and participate in the success of the gaming industry.

1. Research and Development:

Decentralized crowdfunding platforms can support scientific research, technological advancements, and open-source projects. It allows researchers and developers to raise funds for their work, encouraging collaboration and innovation in various fields.

**Chapter 2**

# Literature Review

## Background

This chapter aims to provide information on technologies and concepts used in Crowdfunding dapp. Crowdfunding dapp is based on the Ethereum blockchain. Several aspects make Ethereum a revolutionary blockchain that enables developers to program it to work according to different use cases. Ethereum has various components and mechanism, which make it programmable. It is essential to understand Ethereum before we get any further into our dapp.

### Blockchain and Ethereum

Blockchain is a decentralized and distributed digital ledger that records transactions across multiple computer or nodes. It is designed to be transparent, secure, and tamper-resistance. A blockchain is distributed database or ledger shared among a computer network’s nodes. They are known for their crucial role in cryptocurrency systems for maintaining a secure and decentralized record of transactions, but they are not limited to cryptocurrency uses. Blockchain can be used to make data in any industry immutable.(The term used to describe the inability to be altered).

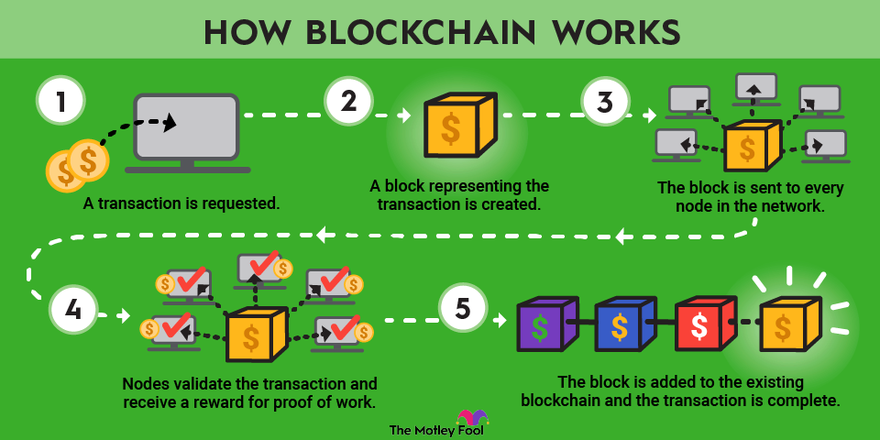


Figure1: About Blockchain

Ethereum ia an open-source public distributed computing platform and operating system based on blockchain technology first used by Bitcoin. Ethereum extends the usefulness on blockchain well beyond cryptocurrencies by making the blockchain programmable according to [Ethereum white paper](https://ethereum.org/en/whitepaper/). It proposed by researcher Vitalik Buterin in his white paper published in the year 2013, where he states the intention of Ethreum is to provide, “a blockchain with a fully fledged Turing-complete programming language that can be used to create “contracts” that are used to encode arbitrary state trasition functions”. These features make it the apt choice for building truly decentralized application similar to this project and many other decentralized applications(dapp).

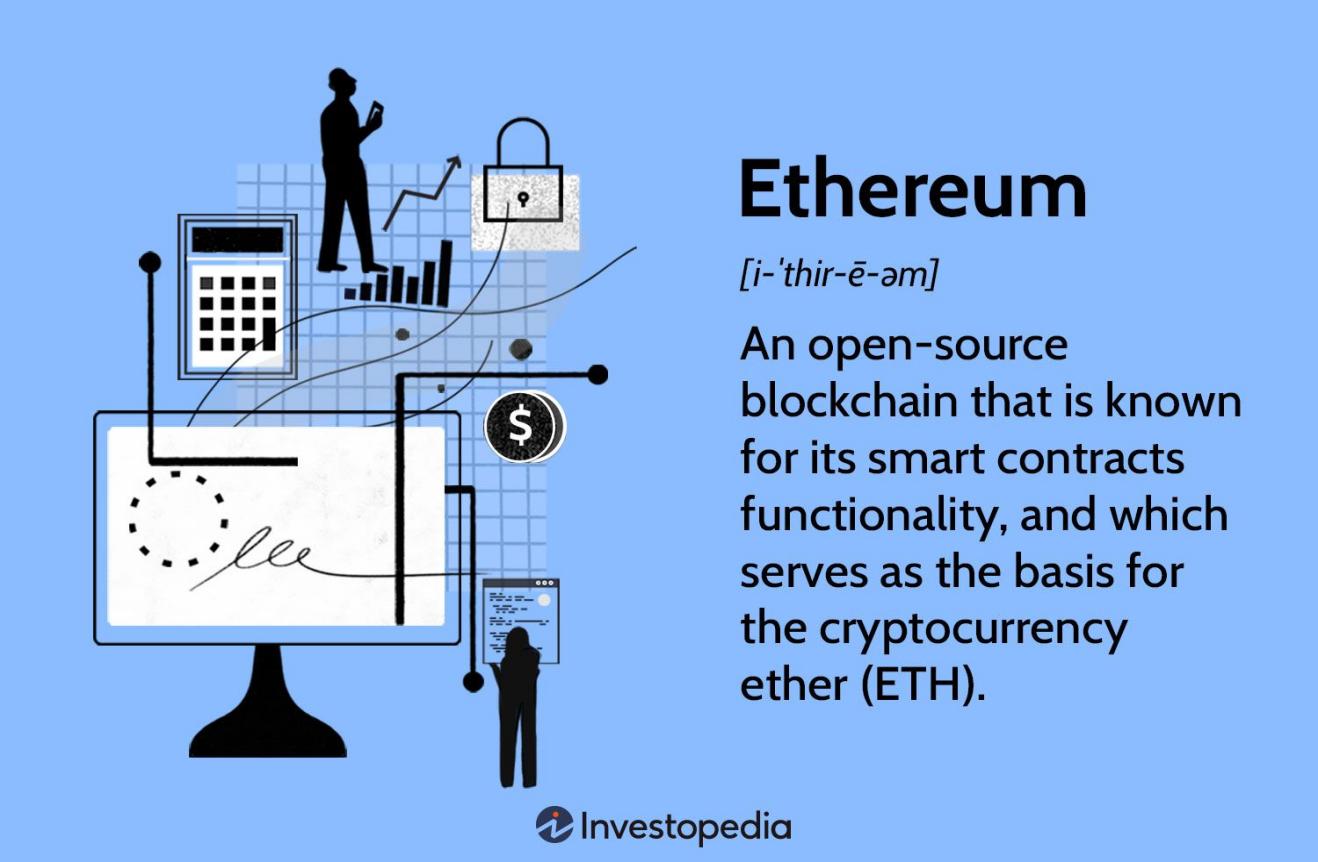


Figure2: Ethereum blockchain

Although Ethereum blockchain is much more advanced and complex, it is still based on the same principle as Bitcoin’s. Ehereum similar to Bitcoin also uses a proof-of-work algorithm (but as of now Ethereum is moving towards the proof-of-stake) run by a peer to peer distributed network to find consensus on the current state of the system, with the miners being rewarded in Ehther(cryptocurrency used by Ethereum network). Network gets transactions from users distributed across the globe and the proof of work algorithm at regular intervals determines a sequence of those transactions to be included in the next block in the blockchain. Every new block added to the chain determines the states of the system. The block creation time in case of Ehereum averages around 14 seconds while that of Bitcoin averages around 10 minutes, both operate on the same set of core principle of blockchain.

The major difference in Ethereum is the complexity of both, the state stored by the blockchain and how the transactions can alter the state of blockchain. Ethereum’s state mainly consists of objects called accounts located by 20-byte address. There are two types of accounts, externally owned accounts(EOA) and contract accounts(CA). EOAs can be accessed by private keys similar to Bitcoin and have a field to store current ether balance of that account. CAs similar to EOAs have additional fields to store the contract code and storage. As CAs are part of contracts, their interaction with other accounts and how they access or modify their storage are controlled by the owning contract. The contract storage is a key value store of data persistent among transactions.

Unlikely externally owned accounts, contract accounts can not initiate transactions on their own. A transaction must have recipient address, a signature identifying the sending account and amount of Ether transferred from sender to recipient. These information is enpugh to move value from one account to another account either in case of CA or COA. However, there exist an optional data field used to specify a public function name present in the contract code of the recipient CA. This data field provides a powerful way for a transaction to induce state changes in the blockchain. Even though contracts con not create transactions they can send “message” to other contracts. Messages are similar to transaction but it takes place between two contracts, by a CA triggering a function in its contract code, rather EOA and CA. Similar to transaction they do contain a message sender, a message recipient, amount of ether to transfer and an optional data field to call a function in the recipient’s contract code.

The actual state changes are done by the Ethereum virtual machine(EVM) upon receiving a transaction, by running the low level contract bytecode. A fascinating concept called gas is used by EVM to operate, it can be thought of as fuel purchased to execute a transaction. Bitcoin also allows scripting for complicated transactions but it is not turing-complete like Ethereum. Ethereum solved a major problem that Bitcoin had by finding a solution to transaction that involve loops. In case a transaction consisted of an infinite loop then calculating the state would be impossible, computational time would be massive, that can create problems for the blockchain. Ethereum’s fairly simple solution to overcome this problem was to have a cost for each transaction that it executed. For every transaction a user creates the amount of gas required for the transaction to be processed is purchased from the ether balance in the sender’s account at an arbitrary price, typically the market price at that moment. Every bytecode operation done in the EVM costs a certain integer amount of gas, operation such as modify or add to contract storage are the most expensive because all those changes are persisted on the blockchain forever. The gas starts depleting when a transaction starts executing and stops if it completes or runs out of gas. If the transaction completes and some gas is left behind then it is refunded back to the transaction sender. However, if it terminates because of gas amount depleted to zero then no ether is returned and transaction fails.

High level smart contract languages used to write contracts, that compile to bytecode executed by the EVM, and make writing a smart contract simple. Solidity , a smart contract language, chosen for this project due to the support available in developer community, object oriented features and syntax similar to JavaScript. Writing a contract in solidity is similar to writing a class in other object oriented programming languages with functions, member variables and interacting with other contracts. The most important feature of a smart contract is their permanence and the immutability of the smart contract code. The access to a function in a contract can be restricted by using modifiers that make it impossible for a developer to call that function if his address does not have the privileges. Even if he adds code to permit his address to call the function, he will not be able to change the contract code once deployed. This might sounds like a burden but there are certain important consideration why this was done. Smart contracts are digital version of written contracts, hence as written contracts can not modified after signed, Ethereum contract code is like a digital contract specifying, with the logical certainty of code, the exact behaviour of the contract account, which can be seen as a kind of autonomous entity on the blockchain. This is required for building applications that are based on completely trustless model where eve the developer can not manipulate once it is deployed, especially when the application involves money. An initial Ethereum dapp that took advantages of the complete trustless model was “The DAO”(decentralized autonomous system). It was all built using smart contracts that users could invest in, make proposal, cast vote on the proposal based on the investments, get rewarded based on the weight of the investments when a proposal succeeds and lot more. It was a widely successful app rising $150 million from user investments, but it was hacked in mid 2016 and lost $50 million of the investments due to a flaw in the code. So no matter how well intentioned a smart contract is, if it is badly coded then it is quite similar to poorly written hand contract and can be exploited by hacker.

Ethereum was the perfect choice for this project “Crowdfunding app” that creates an environment for people to raise money for their dream project. With the smart contracts present, the application con store and manage user account data, store and manage user account data, store metadata and users can interact with the smart contracts using transactions created by project’s client application. Ethereum makes the smart contracts always available, accessible anywhere and restrict control to people who are only authorized to use them. It can achieve this because of the underlying technology “blockchain” first used in Bitcoin.

**Chapter 3**

# Methodology



## Overview

The crowdfunding dapp’s purpose is to create an ecosystem where it will be effortless for everyone to start or support a campaign aiming to create new products or services. Crowdfunding dapp provides a crowdsourcing platform to raise capital by giving well explained project/campaign.

Innovative Teams

Startups

Entrepreneurs

Artists

Social work

Donors

People with passion for donation

Create campaigns and get funding

Contribute to campaigns

Figure: 3 Crowdfunding dapp platform ideology

As seen in Figure3 Crowdfunding dapp platform ideology by using crowdfunding dapp platform teams or individuals such as startups founders, business project leaders, artists, social work campaigns etc. Will be able to raise fund by uploading campaigns and explain them in briefly. The campaign funding is governed by smart contracts deployed on Ethreum. The metadata of a campaign like title, description, image etc are stored on blockchain. Throughout this report, we use two actors “creator” and “donor” who use that platform for their needs, if we are not differentiating the roles we use the term “user” to represent both of them. We use the term “creator” to address a person or a team that want to raise funds. We use “investor/donor” to address an individual who uses the platform to search for interesting campaigns and donate to those campaigns.

To create campaign, a creator has to have clear picture about what he/she wants to build and how the investors who comes to the platform are benefited by this/her product or campaign. If creator himself is not sure about his/her product/campaign then he/she can not expect this/her campaign to succeed. Once he/she finalize his/her proposal, sure about the timelines on his deliverables, has proper resources to build the project and has done some research then the chances of his/her campaign being funded will be hign. All campaigns are timed for raising funds and this time is decided by the campaign creator so as to give maximum flexibility to the creator.

For creating a campaign, a campaign title, description, image is required. Then he/she has to decide time duration for campaign and if he/she get any fund will directly transferred to account of creator. If the donoe comes to dapp he/she will see all campaigns that are asking for funds and if donor interested in any campaign he/she can donate any amount of ethers to creator’s address.

The ideology behind the UI is to keep it as simple as possible for both the investor and creator to use the platform and achieve their goals. Below are the main core UI functionalities that are required for utilizing the platform productively:

- Provide an interface to list all the campaigns.

- provide an interface for donors to donate into a campaign.

- Provide an interface to create a campaign.

## Campaign creation and management

Users must be able to create a campaign and track the donors who have donated ether in their campaign. User lands on the “Register campaign” page and fills in all the details such as campaign title, description, image etc. Then the contract creation is triggred and values provided are used to create the campaign on the Ethreum blockchain.

The campaigns are designed to end according to the time specified by the creator in days. In the campaign creators page all the campaigns are displayed.

### Donating Ether into campaigns

### Users, after getting into a campaign description page can read the campaign and donate Ether into the campaign. After selecting a campaign user can able to donate any amount of fund to the project.

**Chapter 4**

# Development Tools and Planning



## Solidity

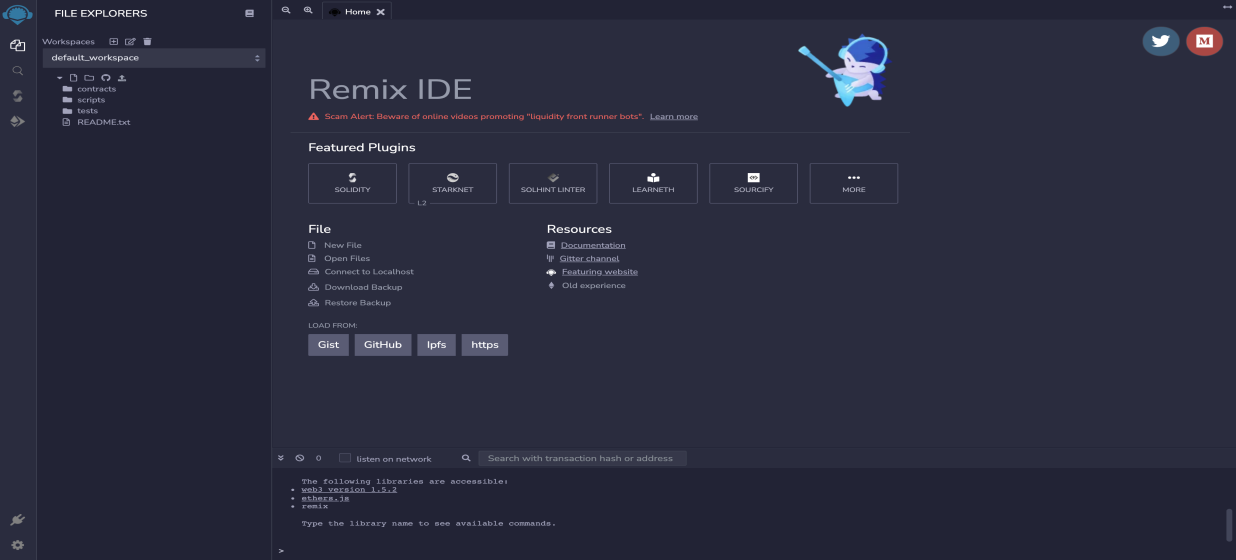
Solidity is an object orientd hign-level language used for writing smart contracts in the crowdfunding dapp. A statically typed language has features such as inheritance, libraries and complex user defined types. C++, python and JavaScript majorly influenced it. Solidity is compiled to bytecode that is executable by EVM. Using solidity developers can write dapps that implement selr-enforcing business logic contained in smart contract, leaving an undeniable and permanent record of transactions.

## Remix

Remix is an integrated development environment (IDE) used for developing snart contracts in solidity. Developing in remix assists develpoers to find bugs and degub code with ease. Remix supports different kinds of environments to deploy and run the smart contracts :

- Remix VM (Sganghai) : It creates a mock of blockchain environment so you can test your smart contract functionality.

- Injected provider (Metamsk) : This environment uses a browser plugin or blockchain based browser to connect to any Ethereum network.



* 1. **Visal studio code**

Visual studio code is a code editor developed by Microsoft. It is feature rich editor that supports code highlighting, code debigging, intelligent code completion and code refactoring to build web application. It supports wide range of programming language and has a built in terminal to execute command line commands. As crowdfunding is a frontend, heavy application with lots of JavaScript Visual studio code was a best choice as the editor to develop the application.

* 1. **Metamask**

Metamask is a chrome browser plugin that acts as a bridge between your browser and Ethereum blockchain by providing a secure identity vault, a user interface to manage multiple Ethreum wallets and sign blockchain transactions. It is one of the best ways to send transactions to Ethereum blockchain because it keeps a track of transaction execution and return if any error occurs during mining or execution. It is an Ethereum community open source project having more than million active users. Hence, it is the most popular plugin to interact with blockchain.

* 1. **Testnet**

Test network is a copy of Ethereum blockchain identical in every way to main network except the fact that their Ether is worthless. There are three types of testnets public, private and GanacheCLI. As name suggests, public testnets are available to everyone and connected to the internet, private testnets are similar to one’s own blockchain and GanacheCLI is a simulation of Ethereum network on a single computer. For this project, we are using a public testnet called Sepolia.

Using metamsk we can connect to any Ethreum network available in metamask. Of all the theree networks Sepolia resembles to the mainnet the most.

* 1. **IPFS**

IPFS stands for interplanetary file system. It is a protocol and network used to store and share hypermedia in a distributed file system. It is an open source project maintained by huge community of developers. The contents in IPFS are accessed in two ways, via FUSE and over HTTP even though IPFS wants to replace HTTP. IPFS can be seen as a BitTorrent swarm, exchanging objects within a single git repository.

## System Requirement

To run our crowdfunding dapp user have computer/mobile that have browsers like chrome, Firefox and in case of software user have to install metamask wallet plugin in their machine.

## Work Schedule

As of beginning we clear about our crowdfunding dapp development process and divided different timeline to develop dapp.

Figure 4: Gantt Chart

We have following timeline to develop our crowdfunding dapp :

- May 10 To July 30 : Research, documentation development, proposal writing etc

- May 12 To May 18 : Smart contract development, documentation

- May 15 To July 15 : Deploy smart contract to testnet, front end file setup, development of front end begin

- May 30 To July 25 : Merging frontend and smart contract to develop crowdfunding dapp.

- Till July 30 : Testing of dapp.

## Expected Outcome

# References

<https://ethereum.org/en/whitepaper/>

<https://remix.ethereum.org/>

<https://ipfs.tech/>

<https://docs.soliditylang.org/>