
Project Topic: Crowdfunding using Blockchain.

Team members:-

Prajwal Deshpande (B-24)

project guide:-

PROF. SANGEETA

Synopsis-

Initially, blockchain was only used as a foundation of cryptocurrency, but today, we can see the rise of this new emerging technology being implemented in many industries. In the future, most technologies around the world are expected to use blockchain as an efficient way to make online transactions. One of the areas in which blockchain technologies can be applied is crowdfunding platforms. The most common problem with the current crowdfunding scene around the world is that the campaigns are not regulated and some of the crowd-funding campaigns turned out to be fraud. Besides, the completion of some projects also was significantly delayed. This project aims to solve these problems by applying Ethereum smart contracts to the crowdfunding site so that the contracts will be fully automatically executed, thus preventing fraud and ensuring that the projects can be delivered within the given duration.

Introduction-

Crowdfunding, to put it simply, is the practice of a group of individuals raising money for a project or campaign without the aid of established institutions like banks or loan providers. Crowdfunding, according to Freedman and Nutting, is a technique for gathering numerous modest donations over an internet platform. Fundraising tool to help fund or capitalise on a well-known business. The crowdfunding campaign consisted primarily of the contributors, the crowdfunding platform, and the project administrators are the three parties. Some common most popular crowdfunding websites are kickstarter.com, indiegogo.com, and mystartr.com.

Project Objectives:

- To study the nature of crowdfunding.
- To determine the role of technology in crowdfunding.
- To analyse various case studies.
- To provide a remedy to address the problems raised.

What are the constraints of the present platforms for crowdfunding?

What advantages can blockchain technology offer for overcoming obstacles?


Reduce the number of clicks a user takes to reach

to make sure our contributors are able to satisfy their needs

to offer an improved user interface

Design - F_end:


We know that the design is more than how something looks, it makes people feel, it makes people act, and it makes a good product great. but for all the good design out there the design process is often chaotic. so we adopted Figma. Figma is a platform where teams design together it unites everyone in your creative process and brings that idea to life in a design wireframe or prototype Figma does it all in the cloud, even if they live on the other side of the globe Figma makes it possible, which enable our team to work together in real-time.

A client-side application that the browser fetches from the host server and renders to the user. Usually written in **NextJS**  **React**, and any other trendy framework.

B_end:

- **Backend** — a server-side application (in some cases — serverless, which means stored in the cloud). Usually, an application takes over all the heavy logic including managing the database. It is interacting with the client side through the API and is written using one of the popular backend languages: NodeJs, and solidity.


- **Backend Layer in Web 3.0 DApp:**


- The major difference between DApp is decentralisation. You don't need a centralised database or web server, you take advantage of blockchain to distribute the application between the network of computers. The blockchain will act as a “state machine”  — maintaining the program state and stability by validating the predefined rules. The state machine is spread between all the participants of a blockchain network and validation is archived through the consensus about program stability.
- The backend logic is implemented via smart contracts that will be later deployed to shared state machines (blockchain network). So backend resides in a peer-to-peer network that everyone can contribute by matching some criteria (ex. owning and locking cryptocurrency in order to vote the changes or make a proposal to the program). But how does the front end work then?

Front-end Layer Communication.

In Web 3.0 DApp

The main frontend architecture in the case of DApp is focused on communication with smart contracts (decentralised programs). And it will differ from the common frontend-backend. Every node in the blockchain network is carrying the state of the program, and if you want to communicate with the smart contract, you have to communicate with one of the nodes, it could be:

- 
- Third-party node providers, like [Infura](#), [Alchemy](#), and [Quicknode](#)
 - Or set up your own node by running the state machine yourself;



Launching the blockchain infrastructure may be quite challenging, especially when you want to scale it to add more nodes. There is when a user provides will be handy, the downside is it creates a centralised dependency component.

All providers are implementing JSON-RPC specifications to communicate with the blockchain networks. RPC, or Remote Procedure Call — is a request-response protocol that defines the rules to allow the client to send the message to the remote machine to execute the function and retrieve the response. In such a communication, the program runs as it is located

on the client's machine, meaning the client doesn't know about the remote machine. All the communication is going through HTTP or Web Sockets.

When providers are connected to the blockchain, the client is able to get information about the state of the blockchain. But how do clients write to The blockchain network? All the write request transactions need to be signed with the client's private key. Each transaction will cost a client a gas fee is a cryptocurrency that will go to appreciate other nodes (miners) which verify the transactions.

Such a provider, like [Metamask](#), is acting like a transaction signer and provider. It stores the private key in the browser and signs when the client makes a transaction request, it also connects to the blockchain network.

Data Layer in Web 3.0 DApp

💡 *It is your data, just Own it!*

Storing the data in blockchain could be quite costly due to transaction gas fees, and it is more reasonable to use non-blockchain solutions like peer-to-peer storage — [IPFS](#) or [Swarm](#).

[IPFS](#) — is a peer-to-peer file system protocol that allows you to store data across a network of machines. Without native support of popular browsers, IPFT uses [private](#) or [public](#) gateways to connect to the network.

Swarm — is similar to IPFT, with the only difference being that the system sustains through the smart contracts in the Ethereum blockchain network.

In order to make your frontend be decentralised as well, you may use the same peer-to-peer storage solutions.