Decentralised Marketplace

Shailesh Mishra

Indian Institute of Technology, Kharagpur, West Bengal, 721302

Prof. Uday B. Desai

Indian Institute of Technology, Hyderabad, Telangana, 502285

Table of Contents

Decentralised Marketplace

Shailesh Mishra

Indian Institute of Technology, Kharagpur, West Bengal, 721302

Prof. Uday B. Desai

Indian Institute of Technology, Hyderabad, Telangana, 502285

Abstract

Blockchain is surely the next big thing. What makes blockchain special is its “decentralised” nature. Blockchain can be used for removing the third party in any field. Hence, it can have a major impact on Marketplace. There are huge firms involved in the digital marketplace, which simply act as a mediator between buyers and sellers. We are not only paying a significant amount for each buy but also, giving an enormous amount of information about us for free to them, without any knowledge. They very well know our likes and dislikes. With the amount of money and data they have, they are probably controlling everything. This is where Blockchain comes in handy by removing the “Middleman”. But, data storage in Blockchain is way more costly than we can think of. Here, IPFS comes to aid, which is a peer-to-peer file system. IPFS is like the sister of Blockchain and if both are combined, we can reach unimaginable heights. This paper is about a small-scale decentralised marketplace using IPFS and Blockchain. It is a two-way marketplace. A seller can put the things he/she wants to sell (with all the specifications) and a buyer can put his wants in it, thus making the marketplace user-friendly. All the details of "wants" and "haves" will be stored in the IPFS. Then, what’s the use of blockchain here? IPFS converts the whole of the file into a hash, which is very difficult to understand for anyone. Hence, Blockchain will be the one storing the goods’ names and their corresponding hash as normal transactions. Blockchain makes this data tamper-proof, hence keeping a good status of the whole of the marketplace. If someone wants to see any item, then he can use the hash, download the file and see the whole of the file. If a person likes the item, he can approach the seller straightaway and buy the item. Thus, there is a significant reduction in price in buying an item, without any breach of privacy.

Keywords: IPFS, blockchain

Abbreviations

Here are the abbreviations that have been used in the paper:

Abbreviations

IPFS	Inter Planetary File System
DAPP	Decentralised Application
USD	United States Dollars
MB	MegaBytes
PC	Personal Computer
p2p	peer-to-peer
geth	Go-Ethereum
wrt	with respect to

1 INTRODUCTION

1.1 Background/Rationale

1.1.1 Current Scenario of Internet

In this age, it's impossible to imagine a day without internet. May it be social media, transferring money, acquiring information or buying things internet always comes to the aid. What makes internet special is the way it works. Let's have a look at its working. We type in something in the search bar and on clicking the button, request for some information about the typed data. The data which we want is stored in HUGE computers which are known as "Servers".The big firms in the internet act as the servers. These firms have all the data that an internet-user needs.Hence, they are able to provide such good results. Once, we make the request, the requested data is searched in "Servers". The most relevant results are then

returned to us in our Web Browser. This looks absolutely fine. All the big firms seem to provide us with the results, we want. It doesn't end here. These firms have made internet better than by giving us better recommendations each day. The current scenario of the Internet seems fine.

1.1.2 The Issue

Where is the issue then? First question. How are these big firms able to make our internet experience better than ever? The answer is DATA. Yes, the data about us. We are giving this data to them without any prior knowledge. Everything that we type in the search bar or every link we click, proves to be a source of data, which in turn proves to be a source money for them. Isn't it a breach of privacy? But, then people would say that it isn't causing any trouble because they are using it for making our experience better. But, if we think properly it does. In a way, we are being controlled. We see what they want us to see. This has become the very nature of the internet. It has become "CENTRALISED". They have an enormous data about us. Probably, the big firms know us better than we know ourselves.

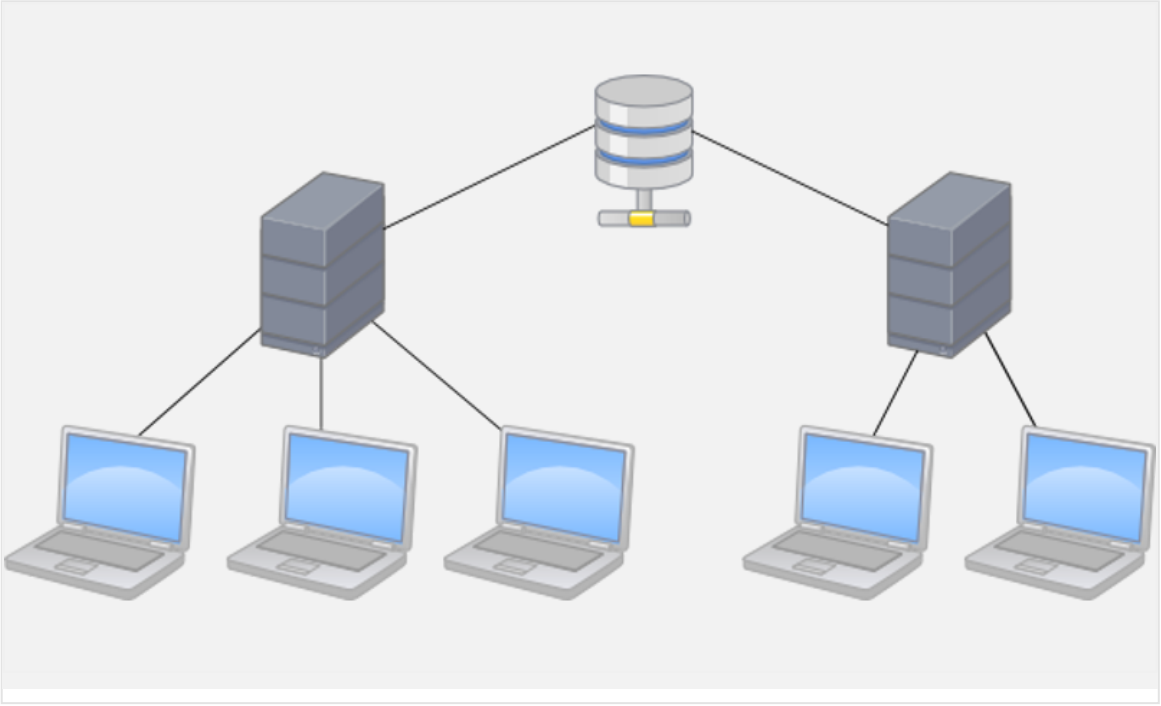

Fig 1 Current state of internet

This doesn't end here. Let's see the issues with "Server". Suppose, you search about "Digital Marketing" and you need it badly. But,there's an issue in the server which contains all the data about Digital Marketing. What's the consequence? You are not able to access the information. The information about "Digital Marketing" cannot be accessed unless the server is corrected. Google Drive and Dropbox are the go-to places for storing files. But, here is an instance that

we can't be very sure while storing files there. Researchers from Northwestern University published a study. It examines how successful teams collaborate using Dropbox. At one point in the study, it says that the company gave them access to project-folder-related data over a two year period from 400,000 users across 1,000 universities. Although Dropbox denied these claims, there's always a possibility that the idea on which one has worked upon for years may be seen by others. Moreover, the servers that store data, are huge bunch of computers that need a huge amount of computational power. They consume an abundance of energy.

Let's get back to the "Centralised" nature of the internet. If they can control what we are seeing, then they can also control what we shouldn't see. Yes, I am talking about Censorship. Censorship is very much necessary in case of adult contents. But, it's a possiblity that vital information may not be shown to us. Let's understand this problem through an example. Wikipedia has been blocked in Turkey since 2017. On 29 April 2017, Turkish authorities blocked online access to all language editions of the online encyclopedia Wikipedia throughout Turkey. This is not a good situation because Wikipedia does hold very important information. Why did this happen? Because the Turkish authorities had to shut down the servers that hosted Wikipedia. Not only this, writers around the world have not been around the world have not allowed to publish their books online.

The drawbacks of the centralised nature of the internet are now very well evident. One of the most visited sites of internet are the e-commerce websites . They have become an integral part of our life. They act as a middleman between us and the seller. They charge both the seller and the buyer for every feature on the internet. This means anyone who is viewing something to buy, he/she is being charged, even if they are not buying. Moreover, they control the things that people see on their websites. Rather, people buy what they(the "Middlemen") want to sell. They have a huge amount of data about everyone. This is not known to everyone. We are giving information about us to them for free without any prior knowledge. Issues like website crash and Mobile App crash must have been also encountered by every internet user. Once the server goes down, we can't use the website for quite a bit of time. This paper concentrates on this part of the internet, i.e., the marketplace and how to make a better overall buying and selling experience.

1.1.3 A possible solution

The word "Centralised" has been used multiple times in the paragraphs above. This is where all the issue lies. So, the remedy is to make the whole thing "Decentralised". What does this mean? This implies that rather than having the Servers(like huge centers) hosting the information, the PCs that are in the network itself become a host for all the information. It sounds really very interesting because a node(a PC in the network) would act as a host and as a client at the same time.

What are the benefits? Acquiring data about someone else will become very difficult because the information that a person accesses will be scattered over the network. Next, the probability of failure of the network is bleak because there will be numerous nodes in the network and multiple nodes can host the same data. There's no way to censor data because it

is distributed among the nodes. So, this type network architecture overcomes all the issues of the centralised for.

One of the system that acts as the same way is "Blockchain". A blockchain is a growing list of records, called blocks, which are connected by cryptographic hash(somewhat like a "Linked List"). The best part of it is that it is resistant to modification of data. There are many technical definitions of Blockchain. But, to put it in simple words "Blockchain" can be used wherever there is a third party involved. Hence, blockchain removes the middle man. It is a p2p network.

Everyone has heard of blockchain in the field of cryptocurrency like Bitcoin, Ether, etc. That is only one of the use cases of blockchain. Like said, before it can be used anywhere there is a middleman. So, the issue of decentralisation can be solved using blockchain. But, there's a glitch. Storing of data in blockchain is way costlier than we can imagine. It will cost about 17,100 USD to store 1 MB of data. At the same time, internet has millions of terabytes, having much less price wrt blockchain. So, the decentralisation does not seem to be feasible.

Here, comes IPFS. It is a p2p file system. The sole purpose of IPFS is to store files. Now, this can be used to store all the files that is provided by the internet. So, will IPFS do the job alone? No, IPFS and Blockchain need to be used together. IPFS produces a hash corresponding to a file and to access that file, the hash has to be known. So, blockchain will act as the ledger for storing the hash of the file. One can use this hash and see the whole file.

So, this whole concept can be very well used in the marketplace, thus helping us to make a decntralised marketplace

1.2 Statement of the Problems

The current online marketplace is completely centralised. Everyone who is using the website is giving his/her data for free. Each of us is being charged for each and everything in the website. The "Middleman" in the marketplace is probably controlling the buyer and seller both. Hence, the need of the hour is the removal of this "Middleman". These all imply that a more transparent and decentralised marketplace is required where there's no breach of privacy.

1.3 Objectives of the Research

This research concentrates on making a DAPP that aims at fulfillinig the following objectives:

1) Decentralising the online marketplace

2) Removal of breach of privacy

3) Making the use of marketplace more transparent

4) Making a base for the decentralisation of internet.

1.4 Scope

The reach of e-commerce websites is huge. Amazon shipped more than 5 billion items in 2017 with Prime worldwide. The reach of online marketplace is very well evident from these numbers. In addition to this, Amazon is not the only e-commerce website running in India. We have Flipkart, Snapdeal, etc. too who are doing really well. So, this DAPP will surely do well if provided the right platform. The thing that makes this application perfect for India is its large population. In addition to this, most of the population is young, which wants technological advancements.

It has been predicted that there would be about 358 million e-commerce website users in India by 2020. So, obviously the reach of this application will be brilliant. It is the solution to the centralised system.

2 LITERATURE REVIEW

2.1 Information

2.1.1 Why not DAPPs everywhere?

Although the idea of decentralised marketplace has been around for quite some time, not many software or technologies have been built for the same cause. There are reasons why DAPPs aren't running in all of our systems. First of all, the technologies like blockchain and IPFS are very complex. These working requires a great deal of understanding. IPFS itself is a very complex system, consisting of many features which are themselves full-fledged systems.

In addition to all of these, blockchain and IPFS require a great deal of computation power to work, which makes them appear expensive when we look at a single PC. But, if we take a network as a whole, a p2p network is cheaper than a centralised one. The issue of cyber attacks also lies in the head of every user while using DAPPs because the PC of one user is directly connected to the PC of every other user in the network. Moreover, IPFS is a technology that works on the assumption that there's a "voluntary-collaboration" among all

the users. This makes the nodes connected in IPFS vulnerable to attacks. Because of all these factors, blockchain and IPFS are not found in most of the PCs in the world, even after having such potential.

2.1.2 Openbazaar

Hence, we haven't seen many DAPPs made for marketplaces. But, there's one that is doing a good job - Openbazaar. This aims at being a fully decentralised marketplace based on IPFS and blockchain. Here, the transactions are carried out through cryptocurrencies. This does make all the transaction more secure. In Openbazaar, sellers can post the items that they want to sell and the buyers can go and have a look at them, just like all the e-commerce websites that we are using. Openbazaar has some very good features. It uses "Escrow" for payments which makes it very user-friendly. Escrow makes the transaction reliable. It also provides an option of "moderator", who is a third party who would help in resolving arguments during the buying of items. The "moderator" can be chosen by the buyer according to the money they charge for moderating. It also has the rating system which makes it easier to find the seller whom we can trust. There's no cost for putting any item for sale. But, the rest all charges are properly shown. This is what is meant by the website being "transparent".

2.1.3 The issues with Openbazaar

This looks perfect. Still, we don't hear everyone using Openbazaar like other e-commerce sites. Openbazaar isn't a household name for every Indian. We can see the idea of p2p marketplace being liked by the people. But, there are issues certainly. First, of all Openbazaar uses cryptocurrency for transactions. The current situation of cryptocurrency in India doesn't look very good. There has been a movement in India to ban cryptocurrency, with a punishment of 10 years jail whosoever uses it. This removes the question of use of Openbazaar. Hence, we need a better method for payment. In addition to this, only sellers can post the items that they want to sell. There should be something where buyers can also post their wants, so that they get items faster. This would make things easier for the sellers too. There are also trust issues on the DAPP too. People are not ready to accept why would a company away from India would make everything free. There's another issue which is, anyone can join Openbazaar. It works on the principle that everyone using the DAPP will collaborate. But, it's obvious that this won't be the case. The glitch lies in the fact that any kind of file can be uploaded into IPFS. Hence, there's a high probability that someone can upload some malicious file. This can harm the whole network.

Hence, if there's a local solution which can make a decentralised marketplace, which can show all the benefits of using a decentralised application over a centralised one and overcomes the above mentioned issues, then we can surely reach a larger mass.

2.1.4 Our solution

The DAPP about which this paper talks about will try to overcome the above mentioned issues. All the relevant data will be stored in files and the files will be uploaded in the IPFS. IPFS produces a hash corresponding to the content of the file. This behaves as the address for the file. Yes, it is like a content-addressed web. So, these hashes will be stored in the blockchain and then, the files can be accessed using the hashes. To fight the issue of malicious data, the DAPP that is being made here is private. It will have some verification method so as to allow only the trusted people to be in the network. Since, this is about a basic version of the DAPP, there's only a very simple verification using a passphrase. This can be surely improved. If this still doesn't work, there will be an option for report. If a person is reported for even once, he/she will be blocked from using the application. For the payment part, the DAPP won't be using any cryptocurrency. The payment will be executed in the same way as the other online payments. But, there'll be a verification. There's a section to post one's wants. This would help the sellers know what to sell. The transparency will be kept same as "Openbazaar".

Blockchain and DAPPs are the future. If we don't start working on these now, India will be far behind. This DAPP will try to overcome all the above-mentioned problems that we have with the current scenario. It will act as a stepping stone for making DAPPs in India and form the base for the new Internet.

2.2 Summary

In this time of 21st century, the concept of peer-to-peer network is being applied in every field. Marketplace involves a large number of people connected and . So, marketplace seems to be one of the best use-cases of DAPPs. But, we can see there's only one such case which is Openbazaar, which is a new term for almost everyone in India. Moreover, its use of cryptocurrency makes it more vulnerable. So, we need a decentralised marketplace which works very well for Indians. So, this paper talks about the basic version of **Deca-ARCADE**, a fully decentralised marketplace that aims at solving all the above-mentioned issues.

3 METHODOLOGY

The major components of "Deca-ARCADE" are:

1)IPFS:

(i) For storing the files consisting the information of the items for sale

(ii) For storing the images of the items that are for sale

(iii) For storing the files consisting whole of the information of the wants

2)Blockchain:

(i) It acts as the back-end of the DAPP.

(ii) Solidity: This is used for writing Smart Contracts which will be executing the codes of back-end.

(iii) Truffle: This is used for making the deployment of the smart contracts easier.

(iv) Ganache: This gave 10 accounts with 100 ethers each for testing of the smart contract(and the DAPP) locally on a PC.

(v) Metamask: This is used for actually testing the whole DAPP in the network.

(vi) geth: This is the core application that helps the nodes to connect to the blockchain.

(vii) The blockchain is completely ethereum-based.

3)ReactJS:

(i) This is used for making the front-end of the DAPP.

4)web3:

(i) This is used for connecting the front-end and the back-end.

After reading all this, the first question that arises is what about deployment of smart contracts which requires cryptocurrency(here Ether). In the previous paragraphs, it was mentioned that the DAPP won't be using cryptocurrency. But, it has been only mentioned that this is the basic version of the DAPP. Therefore, there won't be any mining for cryptocurrency. For testing the DAPP, **metamask faucet**, r**opsten faucet** and **ganache** were used for deploying the smart contracts. Hence, there was no cryptocurrency used.

This section consists of both the use of DAPP in a single PC and in a network. First, the whole DAPP was made in a single PC and then, tested in that PC. Then, this application was taken to a group of PCs and then tested there. The use of the DAPP in a single PC and in a network have only small differences in the back-end. So, here in this section we take a look at the building blocks of the DAPP.

3.1 Account creation

In any marketplace, we have every person has an account. So, here too an account is necessary before accessing the DAPP. While creating the account, basic information of the user is being stored. This information is being stored in a data struture named 'user'.

```
struct user{
    bytes32 name;          //name of the person
    uint id;               //id of the person who is making the account
    bytes32 email ;        //the email id of the person
    bool buyer;            //this is to know if the user will be a buyer
    bool seller;           //this is to know if the user will be a seller
    bool approval;         //if the person has been approved by the head
    uint balance;          //the number of coins the person has
}
```

Fig 2 Data structure that stores basic information about a user

This whole data is being stored in the blockchain. An array, accounts[] of type user is made. This array is the one that stores the account details of each user. The 'id' acts as the unique identity of any user. Along with these data, more information about a user is stored in the multiple '**mapping**' datatype variables.

```
mapping (uint => string) users;                 //corresponding id and username of a user
mapping (uint => bytes32) password;             //corresponding id and password of a user
mapping (uint => address) blockchain_address;   //corresponding id and address in blockchain
mapping (uint => string) ipfs_address;          //corresponding id and ipfs address
```

Fig 3 Mappings to store more data about the user

During account creation, input is taken from the front-end and then, the function addAccounts(arguments) is called which pushes values into the accounts[] and the various mapping variables.

```
function addAccounts(string memory nam, string memory usr, string memory pass,
    string memory em, bool buy, bool sell, address ba, string memory ia, string memory c) public{
    accounts.push(user({
            name: sha256(abi.encode(nam)),
            id: i,
            email: sha256(abi.encode(em)),
            buyer: buy,
            seller: sell,
            approval: true,
            balance: 0
        })
    );
    blockchain_address[i] = ba;
    ipfs_address[i] = ia;
    users[i] = usr;
    get_id[usr] = i;
    i = i+1;
}
```

Fig 4 Function responsible that adds account details in the blockchain

NOTE: sha256(abi.encode(variable)) is used to convert string to bytes32. Using bytes32 in the struct helps in consuming lesser gas while deploying the smart contracts.

The account creation procedure in case of multiple PCs is slightly different from creating multiple accounts in a single PC. The difference lies only in the Blockchain address and IPFS address part.

In case of a single PC, **ganache** is used. The command '**ganache-cli**' gives us **10 accounts** with **100 ethers** each. These 10 accounts provided by it are used to create different accounts in the DAPP. The first account provided by **ganache** is the one that fulfills the gas requirement in the deployment of smart contracts and sending transactions to blockchain. The IPFS address(IPFS peer identity) stored here will be the same for every account because it is unique for a PC

On the other hand, while using the DAPP with multiple PCs, **Metamask** is being used. In each of the PC, Metamask will be added as extension in Chrome, which creates an account with **0 ethers**. **Ropsten faucet** and **Metamask faucet** will be used to obtain sample ethers(no mining) for deployment of smart contract and sending transactions to the blockchain. The IPFS address(IPFS peer identity) is different for all the PCs.

3.2 Putting items for sale

This section talks about how a seller adds his item into the network so that it is up for sale. This section can be divided into two parts:

(i) Storing the information

(ii) Uploading the images and the information in the file system

3.2.1 Storing the information

The basic information that is to be displayed in the website is stored in a data structure **goods**.

```
struct goods{
    bytes32 name;         //Name of the item
    bytes32 brand;        //the brand of the item
    uint s_no;            //the serial number of the item
    bytes32 username;     //the one who is uploading it
    uint selling_price;   //the selling price of the item
}
```

Fig 5 Data structure that stores the basic information about an item

Again here, the s_no as the unique identity for any item that is up for sale. An array, **items[]** is created to store all these basic information about the user. Data is taken from the input and pushed into the array.

```
items.push(goods({
            name: sha256(abi.encode(nam)),
            brand: sha256(abi.encode(br)),
            s_no: s1,
            username: sha256(abi.encode(usr)),
            selling_price: sp
        })
);
```

Fig 6 Adding basic information to the blockchain

Along with this information, the following is also taken from the seller:

(i) Size(if any) (ii) Quantity(number of items) (iii) Age(how old the item is, 0 if new) (iv) Color (v) Other features (vi) Description

The whole of this information is then stored in file of **.csv** format and then downloaded into the PC.

3.2.2 Uploading the images and the information in the file system

Now, the user is instructed how to make the required .zip file. Then, choose this zip file. Then, send it to the network to IPFS. Then, the hash produced by IPFS is stored in the blockchain.

```
//This function stores the ipfs hash of the good
function uploadGoods(string memory a1) public {
    hash1[s1] = a1;
    s1 = s1 + 1;
}
```

Fig 7 Storing the hash of the file in the blockchain

The item is considered to be uploaded in the network only if the folder is uploaded in IPFS. This part is same for both - using the DAPP in single PC or using the DAPP in multiple PCs.

3.3 Putting wants in the network

This section talks about how a seller adds his item into the network so that it is up for sale. This section can be divided into two parts:

(i) Storing the information

(ii) Uploading the information in the file system

3.3.1 Storing the information

Like the goods, the basic information about a user's needs in stored in a data structure - **needs**.

```
struct needs{
    bytes32 name;          //Name of the item
    bytes32 brand;         //the brand of the item
    uint s_no;            //the serial number of the item
    bytes32 username;      //the one who is uploading it
}
```

Fig 8 Data structure to store the basic information of the needs

Again here, the s_no is the unique identity for any item that is up for sale. An array, **requests[]** is created to store all these basic information about the user. Data is taken from the input and pushed into the array.

```
requests.push(needs({
            name: sha256(abi.encode(nam)),
            brand: sha256(abi.encode(br)),
            s_no: s2,
            username: sha256(abi.encode(usr))
        })
);
```

Fig 9 Adding basic information to the blockchain

Along with this information, the following is also taken from the seller:

(i) Size(if any) (ii) Quantity(number of items) (iii) Color (iv) Other features

The whole of this information is then stored in file of **.csv** format and then downloaded into the PC.

3.3.2 Uploading the the information in the file system

Now, the user is instructed how to choose the downloaded .csv file. Then, send it to the network to IPFS. Then, the hash produced by IPFS is stored in the blockchain.

```
//This function is to store the IPFS hash of the file
function uploadNeeds(string memory a1) public{
    hash3[s2] = a1;
    s2 = s2 + 1;
}
```

Fig 10 Storing the hash of the file in blockchain

The need is considered to be uploaded in the network only if the file is uploaded in IPFS. This part is same for both - using the DAPP in single PC or using the DAPP in multiple PCs.

3.4 The list of goods

All the items that have been put for sale, have to be displayed to the buyers in the network. The basic information is acquired from the blockchain and shown in the front-end. These are the functions that return information from the back-end.

```
//To show the name of a good
function sGName(uint z)public view returns(string memory) {
    return name_2[items[z].name];
}

//To show the brand of a good
function sGBrand(uint z) public view returns(string memory) {
    return brand_2[items[z].brand];
}

//To show the username of the good
function sGUser(uint z) public view returns(string memory) {
    return user_2[items[z].username];
}

//To show the selling price of a good
function sGSP(uint z) public view returns(uint) {
    return items[z].selling_price;
}
```

Fig 11 Functions that return basic information about goods

If the buyer finds the basic information appropriate, then the person clicks a button. A link appears at the top. On opening the link, the required file gets downloaded.

3.5 The list of wants

All the requests that have been put up, have to be displayed to the sellers in the network. The basic information is acquired from the blockchain and shown in the front-end. These are the functions that return information from the back-end.

```
//This function is to store the IPFS hash of the file
function uploadNeeds(string memory a1) public{
    hash3[s2] = a1;
    s2 = s2 + 1;
}

//This function returns the name of the thing that you need
function sNName(uint z)public view returns(string memory) {
    return name_3[requests[z].name];
}

//This function returns the brand of the thing you need
function sNBrand(uint z) public view returns(string memory) {
    return brand_3[requests[z].brand];
}

//This function returns the username of the person who has put the need
function sNUser(uint z) public view returns(string memory) {
    return user_3[requests[z].username];
}
```

Fig 12 Functions that return basic information of the requests

If the seller finds the basic information appropriate, then the seller clicks a button. A link appears at the top. On opening the link, the required file gets downloaded.

3.6 Connecting multiple PCs

For this, **ganache** is not used. First of all, **metamask** is to be added as an extension to Chrome. But then the accounts provided by **metamask** has no **ethers**. Then, sample ether is provided to the accounts of metamask in all the PCs that are to be used in the network. All of them are connected to the **Ropsten Test Network**. **geth** is run in all the nodes which makes every node a part of the network. All the nodes are now part of the blockchain. Once a node has IPFS, they are connected. Therefore, the network is ready for the use of the DAPP.

4 RESULTS AND DISCUSSION

In this section, we will look at how the looks.

The homepage is simple and has five sections only. It is just to connect to rest all pages. We will have a look at all the pages.

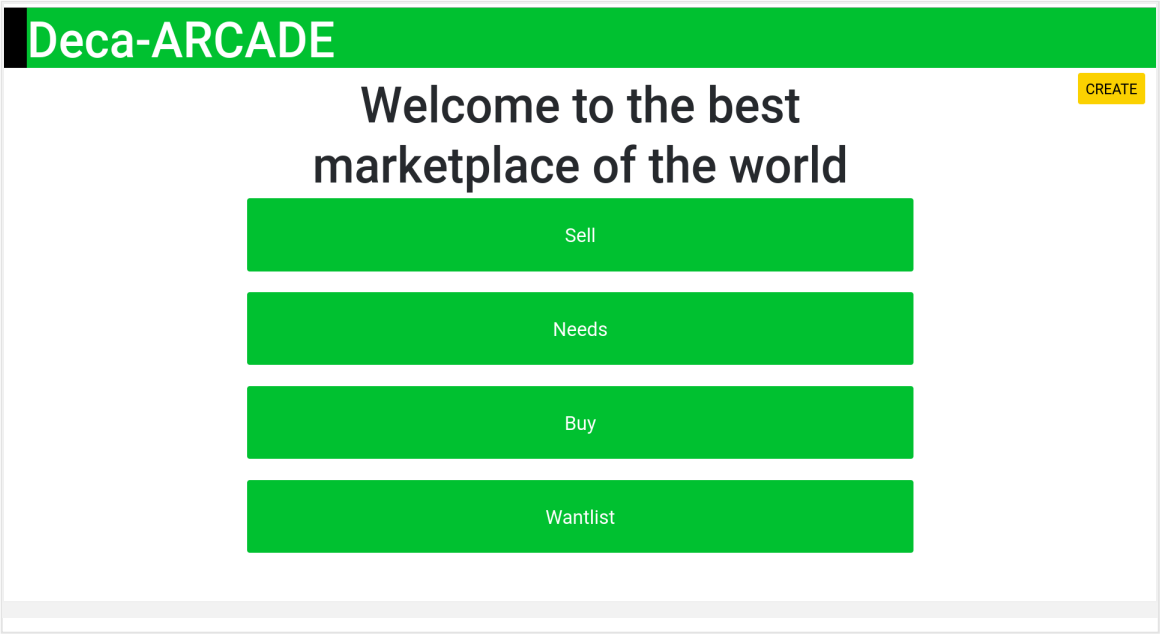

Fig 13 Homepage

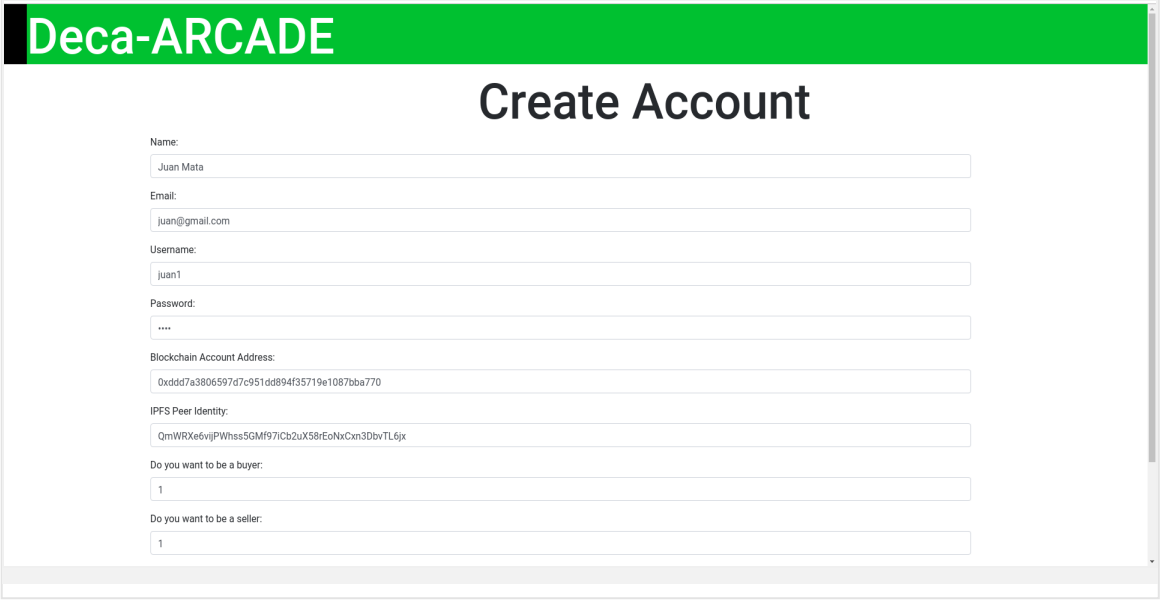

Fig 14 Create Accounts: Taking all the the information about a user and storing it in the blockchain

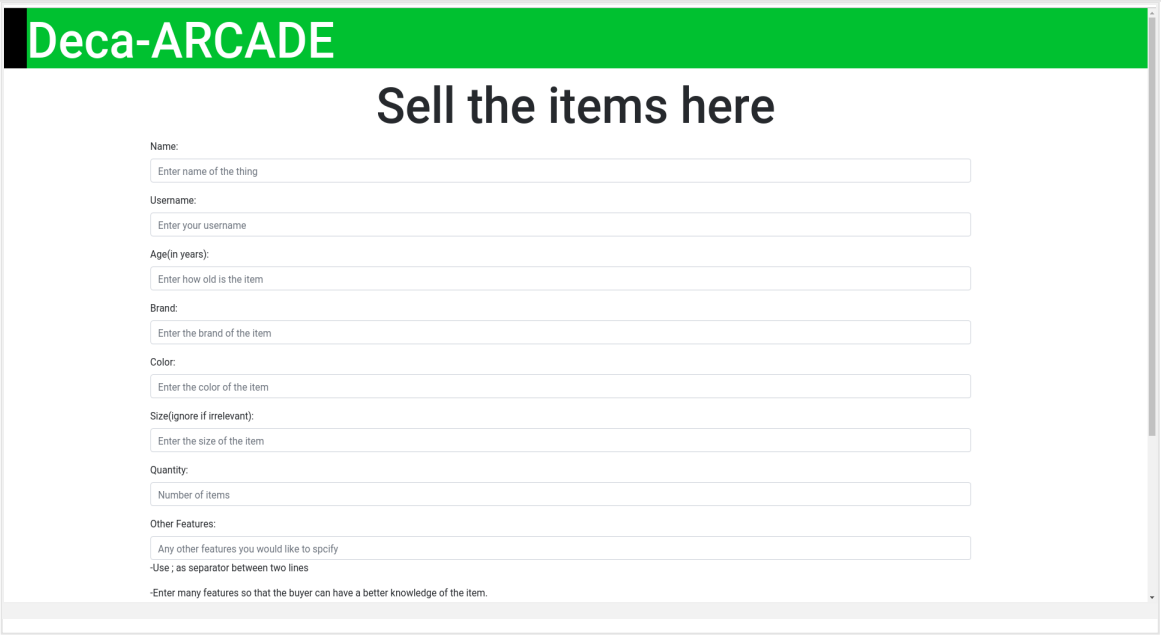

Fig 15 Sell items

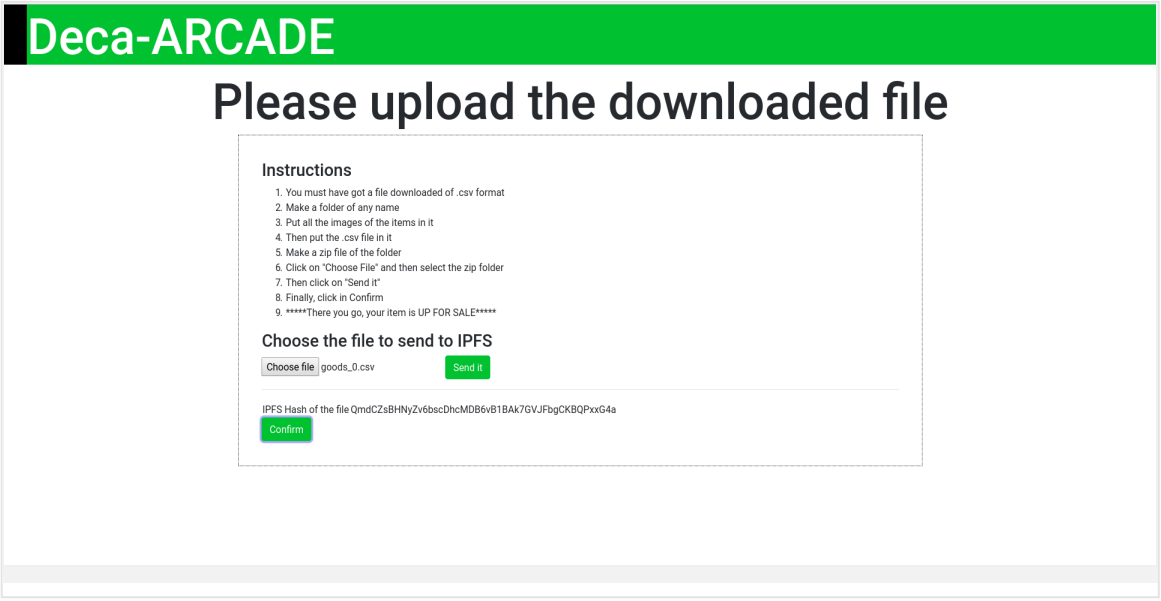

Fig 16 Upload goods in file system

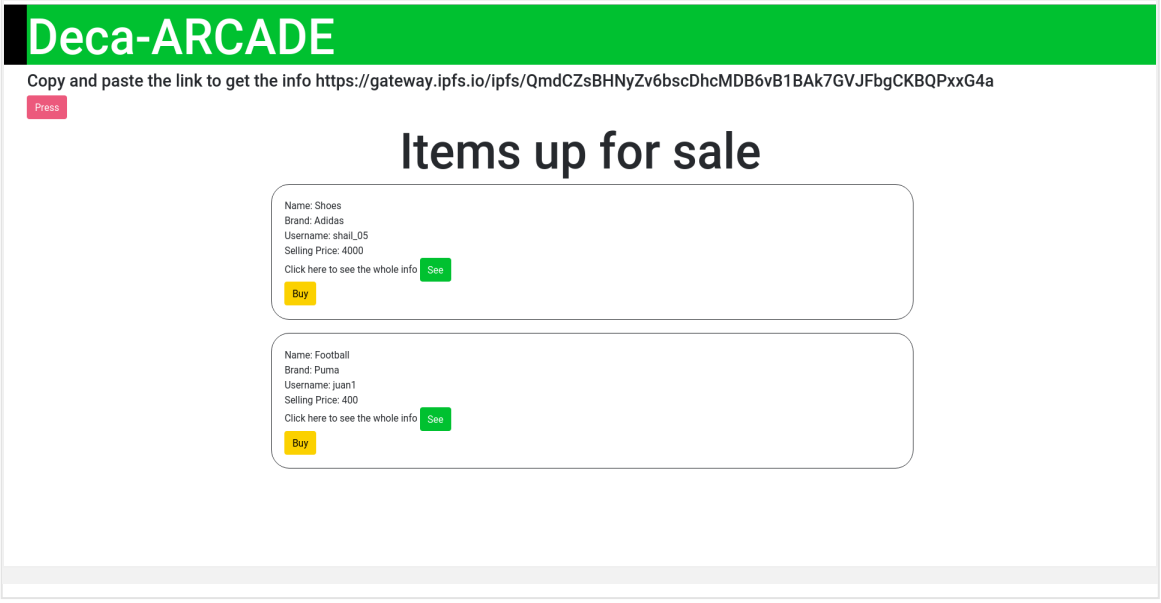

Fig 17 Items list

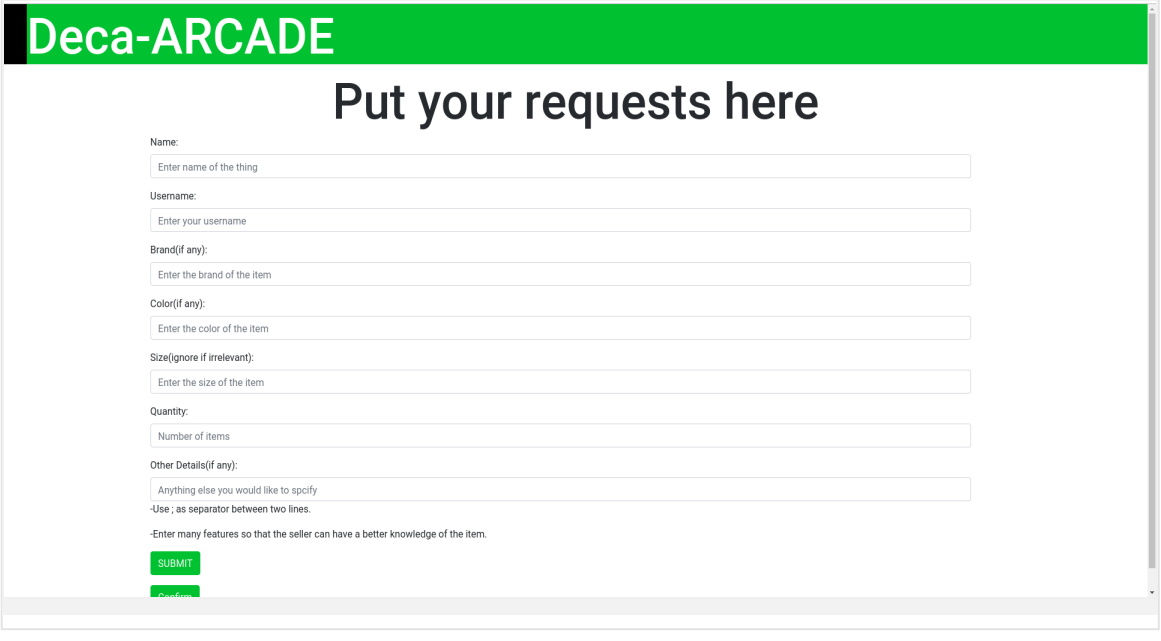

Fig 18 Taking requests as input

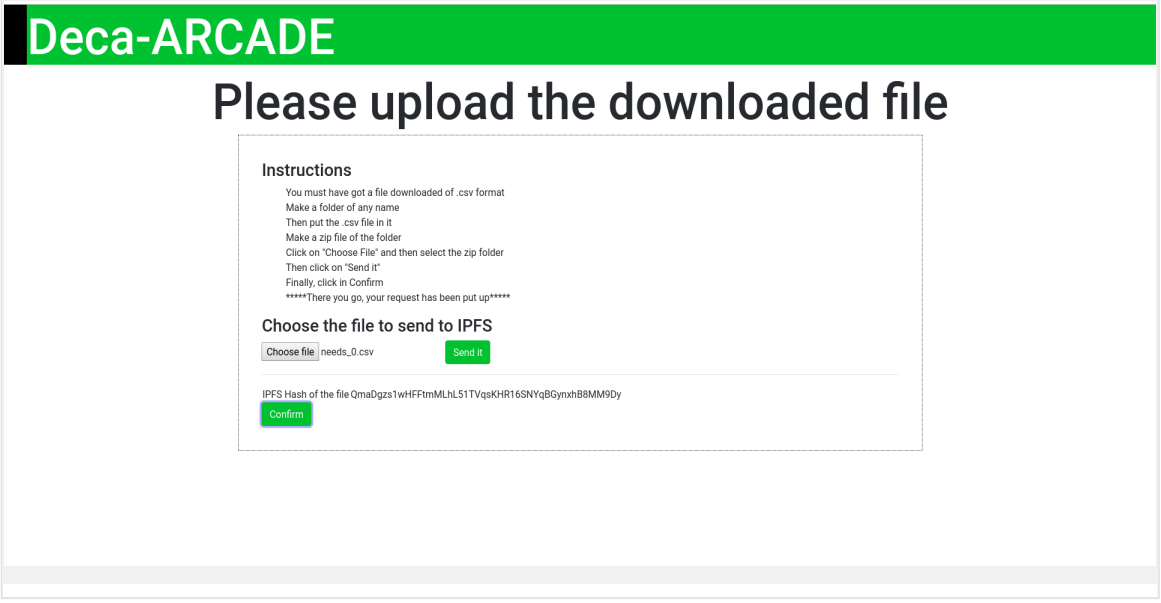

Fig 19 Upload requests into the file system

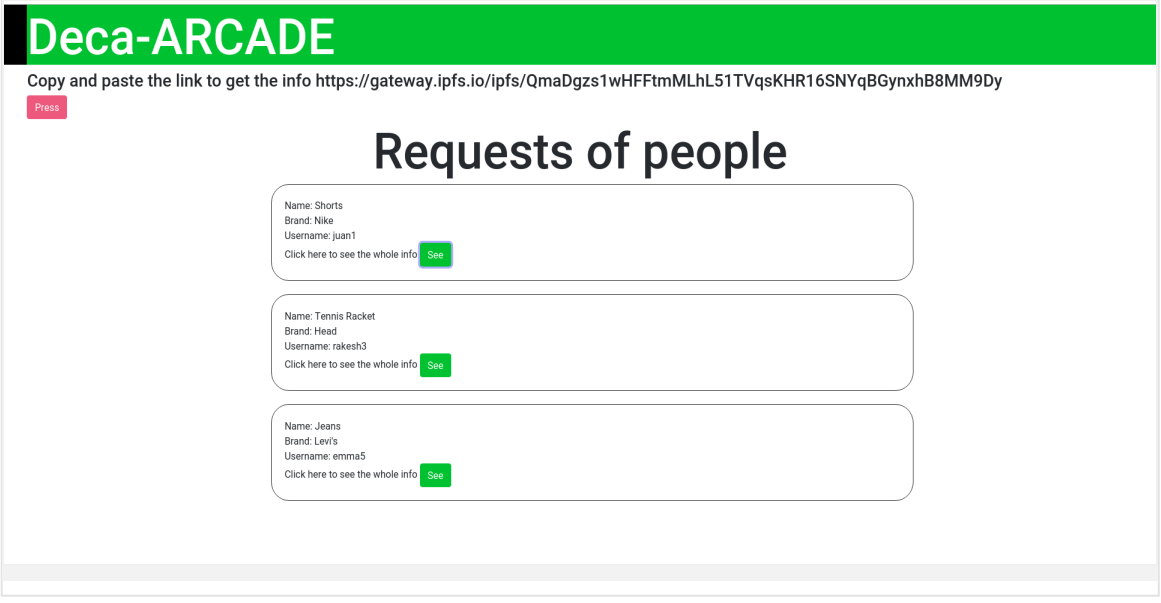

Fig 20 Requests list

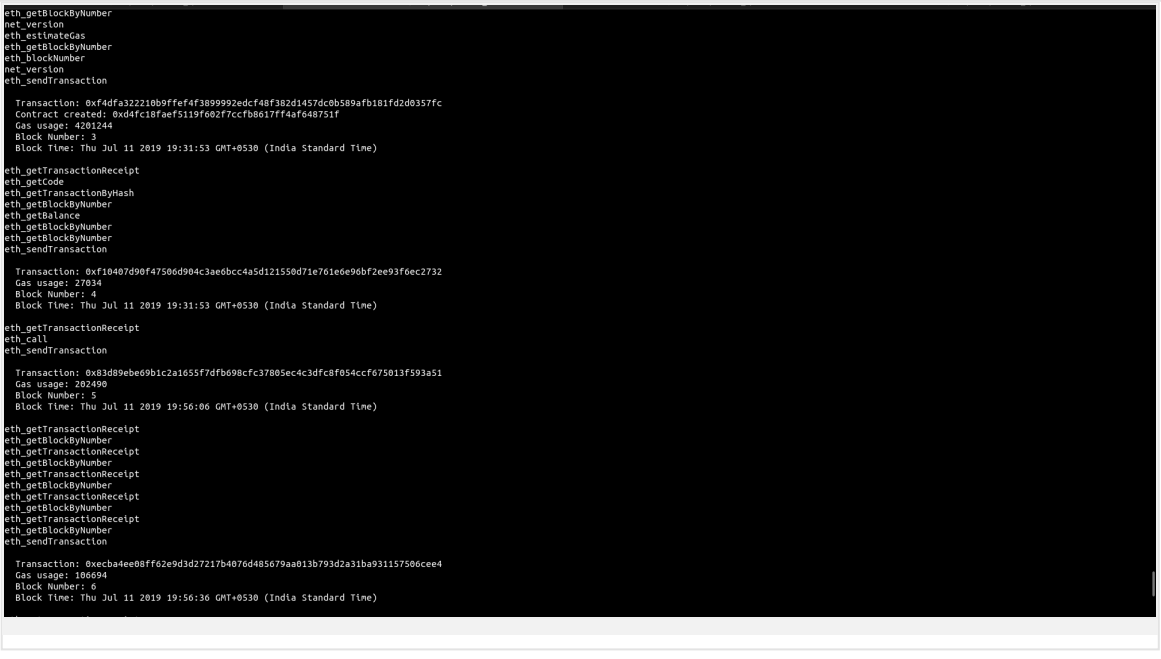

Fig 21 Transactions into the blockchain: As and when the functions are called from the smart contracts, transactions are made and a new block is added to the blockchain

All the pictures here are of multiple accounts in a single PC. Both the blockchain and the IPFS are functioning smoothly. The DAPP is giving desired results in all the sections. The results show that the basic version of the DAPP is ready and it's important to build on this base rapidly.

But, work needs to be done the payment part where we are simply uploading a document regarding the payment. Only then the seller can believe that the payment has been made. This part is not very efficient.

5 CONCLUSION AND RECOMMENDATIONS

DAPPs have not become popular yet. The reason being the performance of the current internet is better than the decentralised one. But the rate of growth of decentralised web is rapid. In this research, there was extensive study of blockchain, IPFS, web3 and ReactJS. These form the backbone of the project. Deca-ARCADE has the ability to become as good as other e-commerce websites, even with additional features. But, a deeper study of IPFS and blockchain is required to make the storage more efficient and make the gas usage less, respectively. Everything has its limitations. Same way, since Deca-ARCADE is in its first version, it has some limitations too. The installation of the packages required is hectic. The DAPP is not yet very user-friendly. A chatting feature will make the DAPP much better. But, it still acts as the starting step for DAPPs. The best part of this DAPP is its scalability and its possible

application in any other part of internet. The aim is to make Deca-ARCADE competent with the other e-commerce websites.

REFERENCES

1. https://solidity.readthedocs.io/en/v0.5.10/

2. https://www.trufflesuite.com/docs/truffle/overview

3. https://www.trufflesuite.com/docs/ganache/overview

4. https://web3js.readthedocs.io/en/1.0/web3-eth.html

5. https://reactjs.org/docs/getting-started.html

6. https://docs.ipfs.io/introduction/overview/

7. https://hackernoon.com/ethereum-development-walkthrough-part-2-truffle-ganache-geth-and-mist-8d6320e12269

8. https://www.eshopworld.com/blog/indian-ecommerce-insights/

9. https://www.forbes.com/sites/louiscolumbus/2018/03/04/10-charts-that-will-change-your-perspective-of-amazon-primes-growth/#684215dc3fee

10. https://medium.com/coinmonks/steps-to-deploy-a-contract-using-metamask-and-truffle-7ae65e6d8dc8

11. https://medium.com/coinmonks/step-by-step-approach-to-create-dapp-using-ethereum-reactjs-ipfs-part-1-42ea4cf69488

12. https://malcoded.com/posts/storing-data-blockchain/

13. https://news.hitb.org/content/dropbox-still-has-questions-answer-after-claims-improper-data-sharing

ACKNOWLEDGEMENTS

I am grateful to the Indian Academy of Sciences for giving me such a wonderful opportunity. A huge thanks to my guide, Prof. U B Desai, for giving me the opportunity to work in the field of blockchain. I would also like to thank him for giving me the freedom to explore more. I would also like to thank Prof. Sathya Peri, for giving me directions constantly during the whole project. Without his guidance, it would have been impossible to complete this project. I would like to thank Mr. Parwat Singh Anjana for helping me out multiple times. Laslty, I would like to thank my family and friends for constant support.

Source

1. Fig 1: https://brutalhack.com/blog/client-server-architecture/