DEGREE/EXPERIENCE CERTIFICATE VERIFICATION SYSTEM USING BLOCKCHAIN

By

Waqas Ahmad

Tahniat Arshed

M.Raees Ahmad

Zahra Uzair Qureshi

Supervisor

Asst. Prof. Ayesha Maqbool, PhD

Submitted to the Faculty of Computing Software Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering

JULY 2020

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis "DEGREE/EXPERIENCE CERTIFICATE VERIFICATION SYSTEM USING BLOCKCHAIN" was carried out by Waqas Ahmad, Tahniat Arshed, M.Raees Ahmad and Zahra Uzair Qureshi under supervision of Dr. Ayesha Maqbool for partial fulfilment of Degree of Bachelor of Computer Software Engineering is correct and approved.

Approved by

____

(Asst. Prof. Ayesha Maqbool, PhD)

Department of Computer Software Engineering

Military College of Signals

National University of Sciences and Technology

Dated: ___ July 2020

ABSTRACT

Blockchain is a new emerging technology which has received great attention in recent years. It is a global online immutable and transparent database which anyone and anywhere with an internet connection can access. The feature that makes it unique is its decentralization. This means the blockchain ledger is shared among all computers around the world, if it is built on public platform, not in one central location. The above discussed features of blockchain i.e. immutability, transparency and decentralization are considered very important in many fields e.g. Maintaining Patient Record, Supply Chain Management, Automotive Industry and Asset Transfer.

This Project, Certificate Verification System is web based Certificate verification service providing one-step verification with multiple options. Verification can be done using student's Digital Certificate by uploading it on the Web portal, generating hash, comparing the hash with the hash of the original certificate stored on blockchain ledger and showing the status. The service can be used by Employers, Placement Consultancies, Colleges and Universities, and other similar organizations. All this process is done by writing a Smart Contract. as this application is being built on a public platform so we used Ethereum.

This page is left intentionally blank.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To Allah belongs the dominion of the heavens and the earth; He creates what he wills. He gives to whom He wills female [children], and He gives to whom He wills males. Or He makes them [both] males and females, and He renders whom He wills barren. Indeed, He is Knowing and Competent.

(Chapter 25: Surah Ash-Shura: Ayat 49-50)

Dedicated to our beloved families and our country Pakistan.

ACKNOWLEDGEMENT

We are grateful to Allah Almighty for giving us strength to keep going on with this project, irrespective of many challenges and troubles.

Next, we are grateful to all our families. Without their consistent support and prayers, a work of this magnitude wouldn’t have been possible.

We are very grateful to our Project Supervisor Dr. Ayesha Maqbool who supervised the project in a very encouraging and helpful manner. As a supervisor, her support and supervision has always been a valuable resource for our project.

Last but not the least special acknowledgement to all the members of this group who tolerated each other throughout the whole year.

This page is left intentionally blank.

Table of Content

1. Chapter 1: Introduction1

1.1 Overview1

1.2 Problem Statement1

1.3 Solution..1

1.4 Approach..2

1.5 Scope..2

1.6 Objectives2

1.7 Deliverables3

2. Chapter 2: Literature Review4

2.1 Introduction to blockchain10

2.1.1 Blockchain and BitcoinError! Bookmark not defined.

2.2 Working Mechanism of Blockchain……………………………………………………………4

2.3 Is Blockchain Secure?...5

2.4 Types of Blockchain………………………………………………………………………………6

2.4.1 Public Blockchain………………………………………………………………………...6

2.4.2 Private Blockchain………………………………………………………………………..7

2.4.3 Federated Blockchain or Consortium Blockchain……………………………………………..7

2.5 Application of Blockchain…………………………………………………………………..8

2.6 Previous Work related to this Project………………………………………………………10

3. Chapter 3: Software Requirement Specification...11

3.1 Introduction...11

3.1.1 Purpose...11

3.1.2 Document Conventions...11

3.1.3 Intended Audience and Reading Suggestions...11

3.1.4 Product Scope..12

3.2 Overall Description...13

3.2.1 Product Perspective..13

3.2.2 Product Function..13

3.2.3 User Classes and Characteristics..15

3.2.4 Operating Environment...15

3.2.5 Design and Implementation Constraints 15
3.2.6 User Documentation 16
3.2.7 Dependencies....................16
3.3 External Interface Requirements....................16
3.3.1 User Interfaces 16
3.3.2 Hardware Interfaces....................19
3.3.3 Software Interfaces 19
3.3.4 Communication Interfaces19
3.4 Non-Functional Requirements 19
3.4.1 Performance Requirements....................19
3.4.2 Safety Requirements....................19
3.4.3 Software Quality Attributes 19
4. Chapter 4: Design and Development 21
4.1 Introduction....................21
4.1.1 Purpose of this Document....................21
4.1.2 Scope of the Development Project 21
4.1.3 Definitions, acronyms, and abbreviations 22
4.1.4 Overview of document....................22
4.2 System Architecture Description 23
4.2.1 Overview of the modules....................23
4.2.2 User Interface Issues....................25
5. Chapter 5: Conclusion....................29
5.1 Purpose....................29
5.2 Limitations....................29
5.3 Recommendations....................29
6. Bibliography....................30
7. Appendix 31

This page is left intentionally blank.

LIST OF ABBREVIATIONS

CLI	Command Line Interface
DApp	Decentralized Application
ETH	Ether (Crypto-currency)
EVM	Ethereum Virtual Machine
GUI	Graphical User Interface
PoW	Proof of Work

[redacted]Chapter 1: Introduction

1.1 Overview

Academic certificates or transcripts are legitimate records given by Universities expressing names, dates of participation, grant of qualification, nationality, grades and so on. It is ordinarily mentioned for by expected managers, offices, colleges, government offices and committees who wish to check whether an individual is a previous understudy of a college or school, the honor and grade they got, and their dates of participation. College understudies far and wide find topographical and authoritative troubles in confirming their records and scholastic archives in different customary manners; in particular via mail, email or face to face. Such a methodology is tedious. Likewise, a significant advance in work enrolling is to experience resumes and candidates' declarations and check in the event that they contain fraudulent or false data.

Be that as it may, because of the nearness of cutting edge and modest examining and printing advances, the imitation of declarations has expanded, which compromises the uprightness of both the authentication holder and the college that has given the endorsement. Subsequently, archive approval and check has become a significant undertaking. It is the way toward guaranteeing that the graduation endorsement introduced by a planned worker is authentic and that the holder is the legitimate proprietor. Additionally, a graduation declaration must be confirmed to guarantee that its substance is valid and furthermore to guarantee that the gave testament originates from a genuine source. As indicated by confirmation is the way toward setting up reality, precision, or legitimacy of something, for example, the verification of official documents.

1.2 Problem Statement

Forgery of Academic Certificates has become a widespread issue in our educational systems, not only creating doubts about the credibility of the institutions but also the educational history of the students. The manual process of degree attestation, in addition, is a time consuming process and not cost effective.

1.3 Solution:

The proposed system is to solve the problem of Counterfeit academic certificates which have been a longstanding issue in the academic community, using the technological blockchain approach which helps in protecting the authentic credential certification and reputation appear.

Our aim is to develop a degree verification system, an online web application that will provide a tamper proof method of verifying certificates, with the convenience of only an internet connection, the user can instantly make an account, login and upload certificates, getting them verified. This will end up saving time and being cost effective. The application will also enable the targeted users to verify an institution certificate online without having to come down to the school. All that is required of the employer is to upload the certificate in order to verify it.

The system improves the speed and quality of services of certificate verification, promotes the globalization of markets and cut down costs. The service will be available for students, employers, placement consultancies, Colleges and universities.

1.4 Approach

To create a degree verification system which is based on the Ethereum blockchain. Ethereum employs a technology called Smart Contracts which will be utilized to store certificate hash on the blockchain, and perform the verification application.

1.5 Scope

The system briefly consists three components in our implementation: uploading and storing of data on blockchain, verification application including federated identity, and the Blockchain. The certificates will be stored in the form of images in a decentralized way using the InterPlanetary File System(IPFS). The InterPlanetary File System is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to identify each file in a global namespace that connects all computing devices. The hash of the image will be saved inside an Ethereum Smart-Contract which will be deployed to our Ethereum network.

The application is designed to merge the hash of the uploaded certificate in a Merkle tree and send the Merkle root to Blockchain, created a record on the Blockchain. The verification application focuses on checking the authenticity and integrity of the certificates that have been issued. It includes one main component: a web-based page. The transaction message is fetched through the blockchain API and compared with the verification data from the receipt. The mechanism can be briefly described in the following way: check the authentication code is valid; check the hash with the local certificate; confirm the hash is in the Merkle tree; ensure the Merkle root is in the blockchain; verify the certificate is not fraudulent.

The blockchain acts as the infrastructure of trust and a distributed database for saving the authentication data. Typically, the authentication data consist of the Merkle root generated using hashed data from thousands of certificates.

1.6 Objectives

During the course of this project, all the aspects of software engineering will be covered i.e. requirement gathering, software design, implementation and testing along with documentation (SRS, SDS, Test Document, Final Report and User manual).

The system, based on the Ethereum blockchain, will provide a tamper proof method of storing certificates and providing verification. Ethereum is an open source, distributed blockchain network that allows decentralized applications to be built on it with the help of Smart Contract functionality.

Through the use of Ethereum smart contract, we can generate and store the hash of academic certificates on the blockchain. This generates a receipt that can be checked at a later date. Tampering is removed because of the structure of the blockchain itself. The blockchain consists of blocks of transaction data. The next block in the chain contains a hash of the previous block, and so any attempted changes to the data will generate the wrong hash and will no longer be considered valid by the other parties in the system. Hence any attempts at editing the certificates will be visible on the network.

Since the certificates will be fraud free, the hash on the blockchain will be legitimate and hence any verifier can validate any copy of the certificate that they possess by generating a hash and comparing it to the hash stored on the Ethereum blockchain.

1.7 Deliverables

Sr.	Tasks	Deliverables
1	Literature Review	Literature Survey
2	Requirements Gathering	SRS Document
3	Application Design	Design Document (SDS)
4	Implementation	Implementation on computer with a live test to show the accuracy and ability of the project
5	Testing	Evaluation plan and test document
6	Training	Deployment Plan
7	Deployment	Complete application along with necessary documentation

1Table 1 - 1

2 Chapter 2: Literature Review

2.1 Introduction to Blockchain

Blockchain has impacted each area and domain of the industry and has been implemented in almost every sector because of its trust building factor in any event occurred. It is distributed ledger that is used to maintain permanent and tamper-proof record of data. It is a decentralized system that excludes the need of any central authority and allows us to trust the outputs of the system. Blockchain provides decentralization because of which the ledger is shared on a public platform and it is not controlled by a single entity. The information is stored in the form of blocks and these blocks are chained together and make a continuously growing database of transactions and are secured cryptographically from any sort of tampering. In short, Blockchain is an open distributed ledger which is used to keep the record of transactions between different parties in a verifiable and permanent way.

2.1.1 Blockchain and Bitcoin

During the past year everyone everywhere went crazy about crypto-currencies and in particular the so-called crypto gold Bitcoin. Underlying technology of Bitcoin is Blockchain. It was introduced by the anonymous person Satoshi Nakamoto who introduced Bitcoin in his white paper “Bitcoin: A Peer-to-Peer Electronic Cash System”[1]. The proposed white paper replaces the need of intermediaries or authorities like banks and financial institutions to facilitate transactions:

“What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.”

Because of bitcoin rather depending upon any third trusted intermediaries like banks to do any action like sending or receiving money or making transaction, it maintains a network of different users which are peer to peer connected and any action or transactions are done on the basis of majority, which in this terminology is called consensus mechanism. The set rules of bitcoin govern how the users in this very network will interact with each other. They characterize:

- The conditions under which a transaction is valid.
- Cost of a Transaction for sending money from one party to another.
- Mechanism of using cryptographic token to validate the transactions.
- Rules on how to change current consensus mechanism.

2.2 Working Mechanism of Blockchain

As already mentioned blockchain is a decentralized and immutable system, this portion will discuss about the working of blockchain i.e. how it is immutable, decentralized and how information gets stored in blocks.

Blockchain is simply a sequence of blocks that are linked with each other. So when the word blockchain is used, it is meant that digital data (in form of block) is stored in public database and are linked(chain) with one another.

When new data is added to this public database it is stored as digital information in the form of blocks. When multiple blocks that store information are joined together then it is called blockchain. To add new block in the ongoing blockchain, however, these four things must occur:

1. There must be some transaction made. Take an example of your online shopping on any website e.g. Amazon. You select a product that you want and then you purchase it.

2. This transaction made for purchasing that product is then needed to be verified. This transaction first must be confirmed by someone and only upon doing so will the purchasing process be completed. Evidently, this means that there must be someone accountable for confirming these transactions or data entries. Using Blockchain technology will make this job much easier. When you buy something from Amazon, the network confirms whether the transaction you made occurred as you wanted it to be or not. Details of the transaction are verified regarding amount, total participants and the time at which the transaction is made.

3. This transaction must have to be stored. Once the transaction is made and verified to be accurate, digital signature of buyer and that of Amazon and the amount of transaction are all stored in a block where it joins hundreds and thousands of others like it.

4. A hash has to be generated for a specific block. And then, once all of the transactions of a block have been verified, a unique number must be given to it, so that the block can be identified. This identification code is called hash. The hash of the previous block is also added in current block and after this it is then added to blockchain.

After addition of the new block in the ongoing blockchain it turns out to be freely accessible for anybody to see and have access to transaction data. Taking example of blockchain of Bitcoin, one has access to transaction information along with data about when, where and by whom he blocks was added to this blockchain.

2.3 Is Blockchain Secure?

Blockchain technology represents the issues of trust and security in a few different ways. Initially, new blocks are directly stored in sequential and chronological order. Because of this after a block is mined it is added at the end of blockchain. Size of each block is 1MB. As an example, the bitcoin blockchain has been growing and till September 2018 it is of size 185GB since 2009 when bitcoin crypto currency was created and first used as virtual money. Blockchain is immutable as nobody can modify data or distributed ledger of all blocks because after creation of new block and to place it at end of whole chain makes it very difficult to go back to some previous block and alter its data. It is also ensured by concept of hashes; each block contain its block hash along with the hash of previous block. Hashes are codes in form of strings that created by complex math functions and algorithms which converts any form of data into a unique alphanumeric string. If information in any of previous block is changed then its hash will also change as well. If somebody alters the information for example in mid of blockchain then he has to change the hashes from mid of block chain to end of blockchain block by block step by step and after it is done this altered blockchain will be entirely different from the one that all the other nodes in the network have on their computers.

Here is for what reason this is critical to security. Let us suppose someone tries to alter your transaction of Amazon so you have to pay for your purchase twice. The hash of block will be changed as soon as the amount is changed. In order to cover this intruder has to change the hash of block and update it but this block contains the hash of the previous block as well. However, in this effort hashes would get changed block by block, and soon.

Thus, in attempt of changing data of a single block the hacker has to change every single block that comes after the block he wants to change. For this he must recalculate all these hashes that demands time and very high computational power. In short once a block is created and added then it is very difficult to edit it and not possible to delete it.

2.4 Types of Blockchains

The thought developed that the Bitcoin blockchain could be utilized for any sort of significant transaction or any sort of understanding, for example, peer to peer protection, trading, insurance and transferability and so on[5]. Colored Coins and Mastercoin attempted to tackle that issue by using Bitcoin Blockchain Protocol. The Ethereum venture decided to produce their own blockchain and deploy it on the main network which will also be used by others who deploy their smart contracts on the Ethereum blockchain.

Institutions like banks realized that the underlying idea of bitcoin can be used as “Distributed Ledger Technology (DLT)” and establish a permissioned blockchain (private or federated), where the validator is from same organization. Thinking of blockchain as a permissioned private ledger is profoundly dubious and controversial. This is the reason the term “Distributed Ledger Technology (DLT)” developed as more general term.

Private Blockchain may not revolutionize the world, however it holds the possibility to supplant most elements of customary money related establishments with programming, in a general sense reshaping the way the budgetary framework works.

Private Blockchain is important for settling proficiency, security and extortion issues inside conventional money related foundations, however just steadily. It's not in all respects likely that private blockchains will reform the monetary framework.

2.4.1 Public Blockchains

Currently the public blockchain is based on protocols like proof of work, consensus algorithm that are easily accessible and open source. Anyone can make their local device a public node and can be part of consensus by validating the ongoing transactions. Consensus algorithm is process that specifies which block is to be added to the blockchain and tells about the current state. Transactions can be made by anyone around the globe and these are then added to blockchain after validation. Anybody can have access and see the transaction on the public blockchain. Examples of public blockchain include Bitcoin, Litecoin, Ethereum, Dodgecoin, etc. Public Blockchain lessens the expenses of making and running "Decentralized application (dApps)" without any need to maintain servers. It has zero infrastructure cost. It has an impact on current business models by intermediation.

2.4.2 Private Blockchains

Private Blockchain is also known as Permissioned Blockchain. In this type of Blockchain, permission is granted to the nodes that want to be a part of the Private Blockchain. This type of Blockchain is built in a case where the owner doesn't want to give read and write access to everyone. In this type of Blockchain, the concept of identity is used. This means that for every node of the network the owner of the Blockchain predefines some set of rules e.g. whether this node can read the data or not. This is possible only if the owner has the identity of all the nodes. Another interesting fact is that in private blockchain, the nodes are known to everyone in the network. If some unwanted activity happens, the whole network is going to know about the party that is attempting to tamper with the data.

2.4.3 Federated Blockchains or Consortium Blockchains

Consortium Blockchains are quite similar to Private Blockchains.We can say that these types of blockchains are semi-private. Unlike Public Blockchains, Consortium Blockchains are used in areas where the owner needs to retain the control and privacy of the organization for example banks. Since these are semi-private blockchain, the rights to mine and add each block on the blockchain through consensus are given to only specific nodes that can validate the blocks. Other nodes don't have the authorization to confirm any action or transaction but they are allowed to read what is happening on this blockchain. These different type of nodes with different rights are defined in predefined set rules.

Consortium Blockchains work under the association of a group. Rather than open Blockchains, they do not give permission to validate transactions to all persons. These are more efficient with higher adaptability and give more transaction security. Federated blockchains are for the most part utilized in the financial area. The consensus mechanism is restricted to certain nodes. Suppose there is bank that uses consortium blockchain and it has 20 branches and every branch is considered as a node. If 10 branches are given rights to verify transaction then other nodes cannot participate in this consensus but they may read the blockchain. All this is defined in smart contract

Examples includes Corda, r3 (Banks), B3i (Insurance), Energy Web Foundation (Energy)

It is still controversial that whether this should be considered as blockchain or not. Blockchain is new and advancing technology and much work is yet to be done that's why it is still not clear how it would be accepted and how will it stand. Many people hold the view that private or consortium blockchain may endure destiny of intranets like in late 90s when private organizations and businesses made their own LANs/WANs that are private to them. This is done so as to oppose the use of public internet and administrations which after the appearance of SAAS in Web2 it seems to be pretty much out of date.

2.5 Applications of Blockchain

2.5.1 Financial Services

The advantages that blockchain provides over traditional systems is that the already existing systems relies on trusted third party or Intermediaries to do the actions or transactions and run the process but using blockchain makes the process transparent, time efficient and cost effective by giving immutability without being error prone, inconvenient and unmanageable.

2.5.2 Asset Management

In Asset Management, Trade processing and settlement can be of high-risk and expensive, especially when it involves two different countries. Since both the parties' buyer and seller keep and manage their own record of data, there can be poor corroboration of data since the records need to be matched time and again and this also increases inefficiencies. Block chain reduces errors in asset management because of immutable data and provenance is easily discernable.

2.5.3 Insurance

In insurance agencies some cases have been seen when person comes to claim insurance with fraud. Insurance agents have to see the whole case to determine if it is valid or not or if the person who claims had used any abandoned policy or some other incorrect method. This whole process is manual, which makes it a tedious task and prone to error. Blockchain prevents false claim by its set rules that use hashes and data cannot be changed once block is added thus giving transparency and immutability. So, providing risk free management, blockchain also save time by making the whole process very easy.

2.5.4 Blockchain Healthcare

Individual health records can be stored on blockchain database, so it could be easily accessible and record management would be much easier. A similar procedure could be utilized to guarantee that exploration and research is directed by means of HIPAA laws (in a safe and classified way). Receipts of all the medical processes carried out can be recorded on a blockchain and can be sent to insurance agencies as proof of delivery. This database can also help for managing general health care records, such as drug supervision and regulatory authorities, to manage and test result and healthcare services.

2.5.5 Blockchain Internet-of-Things(IoT)

When day to day objects are embedded with computerized and digital devices and connected to internet in order to send or receive data, it becomes Internet of Things. When connected to a network a normal object just not only behaves as it is, but it becomes so called smart objects or devices. These objects are now people connected to other people and objects and devices with one another. According to some analysts there will be more than 26 billion IoT devices.

2.5.6 Blockchain Identity

Online companies have all the information about us because we gave them this data. Once we upload anything or give any information on internet it remains there forever, even after it has been deleted from our side. Some online companies from where we buy or purchase something or do any transaction shares or sells our details to others who can then use this data and send us their ads according to known interests or information. The blockchain prevents this by making protected data point from where one can only encrypt the relevant information that is required at certain times. For example, going to bar only needs information about age to be shared to know if he is 21 or not. This way, additional information will not be unnecessarily exposed to third parties.

2.5.7 Blockchain Government

In most of the countries elections and its results are a big issue and most of the time security of the voting system is questioned or sometimes winning party is accused of rigging. This can be prevented by using voting system based on blockchain in which not only the votes will be encrypted but also individuals can cast their vote and confirm it easily. This voting system will be cost and time efficient for both the individuals and the government too.

According to report of McKinsey and Company in 2013 the world would be richer by 2.6 trillion USD if freely accessible open government source data is accessible for all the individuals over the internet. New business ventures can use this information to unmask and expose the fraudulent schemes. It can also be used to investigate the side effects of medicines or used by farmers for precise farm-cropping. In the present scenario this information is released yearly which is largely, non-responsive to citizens input.

The blockchain provides open responsive data to the citizens anytime anywhere as they want. The blockchain, as a public ledger, can open this data to citizens whenever and wherever they want.

2.6 Previous work related to this project

Previous work done on this idea are discussed in this chapter. There were a few projects that were based on the idea of blockchain, and the following is a detailed description of projects previously carried out in this context.

The current technology that exists is a Blockchain based Academic Certificate Authentication System based on cryptographic protocols like multi-signature, BTC Address –State-Based revocation mechanism and trusted federated identity to ensure the legitimacy of the issuing institution. The multi signature scheme ensures that forging of certificates is made to be as difficult as possible, since each issued certificate will need to be signed by a majority of the academic committee members. The system itself consists of four components: the Issuing Application, Verification Application, Blockchain, and local database implemented by MongoDB.

The system is implemented using the Bitcoin Blockchain. The **bitcoin blockchain** is a public ledger that records **bitcoin** transactions. It is implemented as a chain of blocks, each block containing a hash of the previous block up to the genesis block of the chain. The system that we are implementing, however, will utilize Ethereum technology. Ethereum serves as a building platform for dApps and smart contracts, allowing it to send tokens that represent values that do not necessarily need to represent currencies. Ethereum's block time also amounts to seconds, as compared to Bitcoin's which takes minutes. Ethereum has more use cases than Bitcoin. With its smart contract technology, we can build dApps that do not go offline and cannot be altered.

3 Chapter 3: Software Requirement Specification

3.1 Introduction.

In this chapter we are going to discuss purpose of SRS. Its document conventions, who are intended audience. In general this chapter is going to provide overview of the online degree verification system in terms of explanation of what we are going to develop. It also contain some diagram which explains the system even further. It contains class diagram by which one can get brief overview of how the users will interact with the system.

3.1.1 Purpose

This document details the software requirements specification for the Degree/Experience Verification System using Blockchain. It reflects all the requirements, constraints, and design activities of this project. The release number of the software is 1.0.

The idea of the project is to develop a Blockchain based secure system, a technique which is mainly implemented by conflating the hash value of local files to the blockchain, which is an effective technological approach protecting authentic credential certification and reputation. This document is meant to outline the features and requirements of project, to serve as a guide to the developers and a software validation document for the prospective client.

3.1.2 Document Conventions

Italics: The words in italics are further explained in the glossary.

3.1.3 Intended Audience and Reading Suggestions

The Intended audience for this document is listed below:

3.1.3.1 Examiners/Evaluators

The document will provide the FYP evaluators with the scope, requirements and details of the project to be built. It will also be used as basis for the evaluation of the implementation and final project.

3.1.3.2 Developers

The document will provide guidance to the developers to determine what the requirements are and how they should continue with the project.

3.1.3.3 Project Supervisor

This document will be used by the project supervisor to check whether all the requirements have been understood and in the end whether the requirements have been implemented properly and completely.

3.1.3.4 Project Testers

Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. It will help in building up test cases for the testing process. This way testing becomes more methodically organized.

3.1.3.5 Up gradation Engineers

Up gradation engineers can review projects capabilities and more easily understand where their efforts should be targeted to improve or add more features to it. It sets the guidelines for future developments.

3.1.3.6 End Users

This document can be read by the end users if they wish to know what the project is about and what requirements have been fulfilled in this project.

3.1.4 Product Scope

The system briefly consists of the following components: uploading and storing of data on blockchain, verification application including federated identity, and Blockchain.

The certificates will be stored in the form of images in a decentralized way using the InterPlanetary File System(IPFS). The IPFS is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. The hash of the image will be saved inside an Ethereum Smart-Contract which will be deployed to our Ethereum network.

The verification application focuses on checking the authenticity and integrity of the certificates that have been issued in order to avoid fraudulent accreditation. The verification party uploads the certificate or degree to the portal, the system generates a hash of the certificate and compares it to the hash stored on the Ethereum blockchain, and returns a result as to the validity of the certificate.

The blockchain acts as the infrastructure of trust and a distributed database for saving the authentication data. Typically, the authentication data consist of the Merkle root generated using hashed data from thousands of certificates.

3.2 Overall Description

3.2.1 Product Perspective

The proposed system is designed to solve the problem of Counterfeit academic certificates which have been a longstanding issue in the academic community, using the technological blockchain approach which helps in protecting the authentic credential certification.

Based on Blockchain, a series of cryptographic solutions are proposed to resolve the issues above, including, utilizing a multi-signature scheme to ameliorate the authentication of certificates; exerting a safe revocation mechanism to improve the reliability of certificates revocation; establishing a secure federated identification to confirm the identity of the issuing institution.

3.2.2 Product Function

The designed product will perform the following functions. The two major components of the product are:

1. Uploading and Storing Data on Blockchain:

   This application will perform the following operations:

   - Upload the certificate image
   - Upload the image to IPFS
   - Calculate the hash value of the image file
   - Storing the hash to the Ethereum blockchain

2. Verification Applications:

   The verification application will check the authenticity and integrity of the academic certificates that have been issued. In order to perform this, it has to perform the following steps:

   - Upload the PDF files.
   - Calculate the hash value of PDF files.

- The client makes a request with the blockchain to retrieve the required certificate. This is an interaction with the block chain API.
- The logic of the verification is as follows:
  Authentication management: The issuing address can be used to check the relationship with the school identity.
- The hash value on the certificate is verified to avoid tampering.
  The verification to confirm if the hash value is in the Merkle tree.
- The verification to confirm if the hash value of Merkle tree root is on the blockchain.
  The verification of the validity of the certificate (to avoid the cancelled certificate).
- The verification of the valid date of the certificate (to avoid the expired certificate).

Users of the System:

1. Student
2. Checker
3. Issuer

Students:

Students can perform the following functions:

1. Apply for new certificate.
2. Look up the latest certificate progress
3. View and download their issued degree documents.
4. Share the Degree Documents with the Employers.

Checker:

1. Check the applied application.
2. Approve the application and add feedback.
3. Generate the certificate draft for further signs.
4. Compare the student’s information with API’s.

Issuer:

1. Merged the certificates to the transaction.
2. Serialize and sign the transaction using the private key.
3. Approve the certificates revocation request.

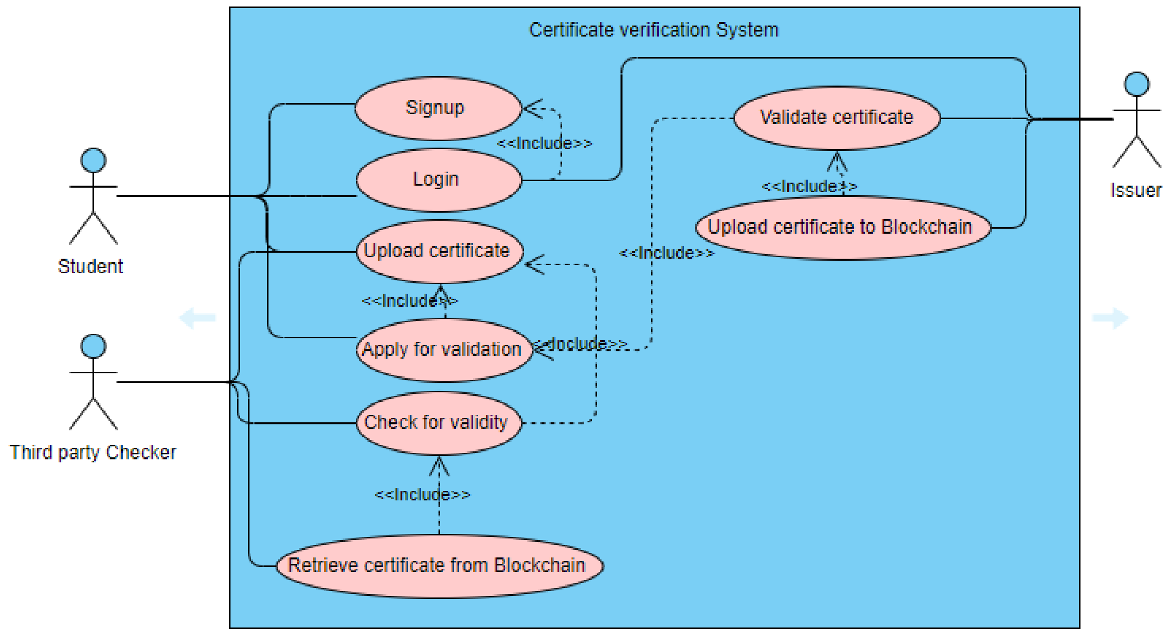
4. View and download the certificate documents.

3.2.3 User Classes and Characteristics

Following are our targeted users:

- Students: they will be responsible for uploading the certificate in order that it may be stored on the blockchain.
- Third party checker: this will be the verification party that will check for the validity of the certificate by uploading the certificate to the web. The system then performs a comparison of the uploaded certificate's hash and the hash of the original document stored on the blockchain.

User and his interaction with the system is shown in Figure A.



2 Figure 3 - 1

3.2.4 Operating Environment

- Linux, Ubuntu 16.10 or 18.04
- Extensions

3.2.5 Design and Implementation Constraints

- The user can only use the predefined graphical elements to create a program
- The language based on the X-man component model
- User can create programs by following the tutorials given in the user manual.

3.2.6 User Documentation

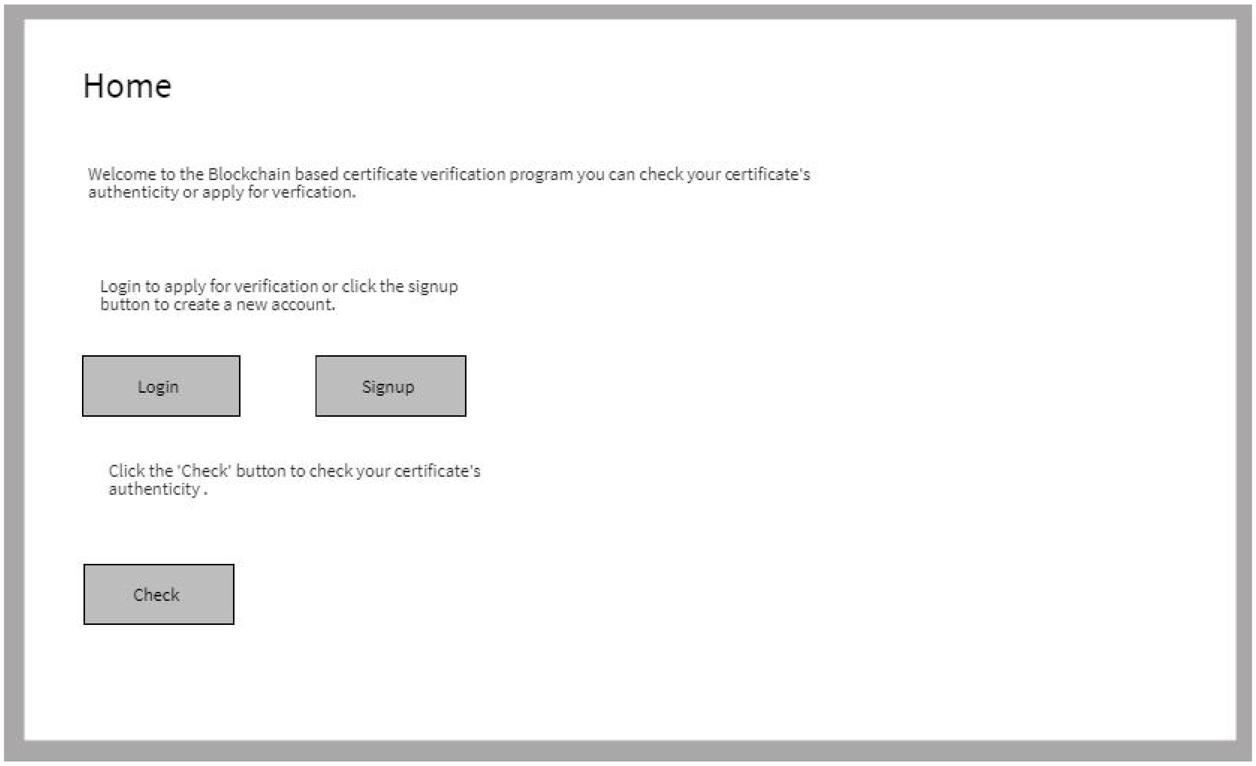
This interface is easy to use and self explanatory and hence there will be no need for user guide.

3.2.7 Dependencies

- Internet Connection
- No other such dependencies

3.3 External Interface Requirements

3.3.1 User Interfaces



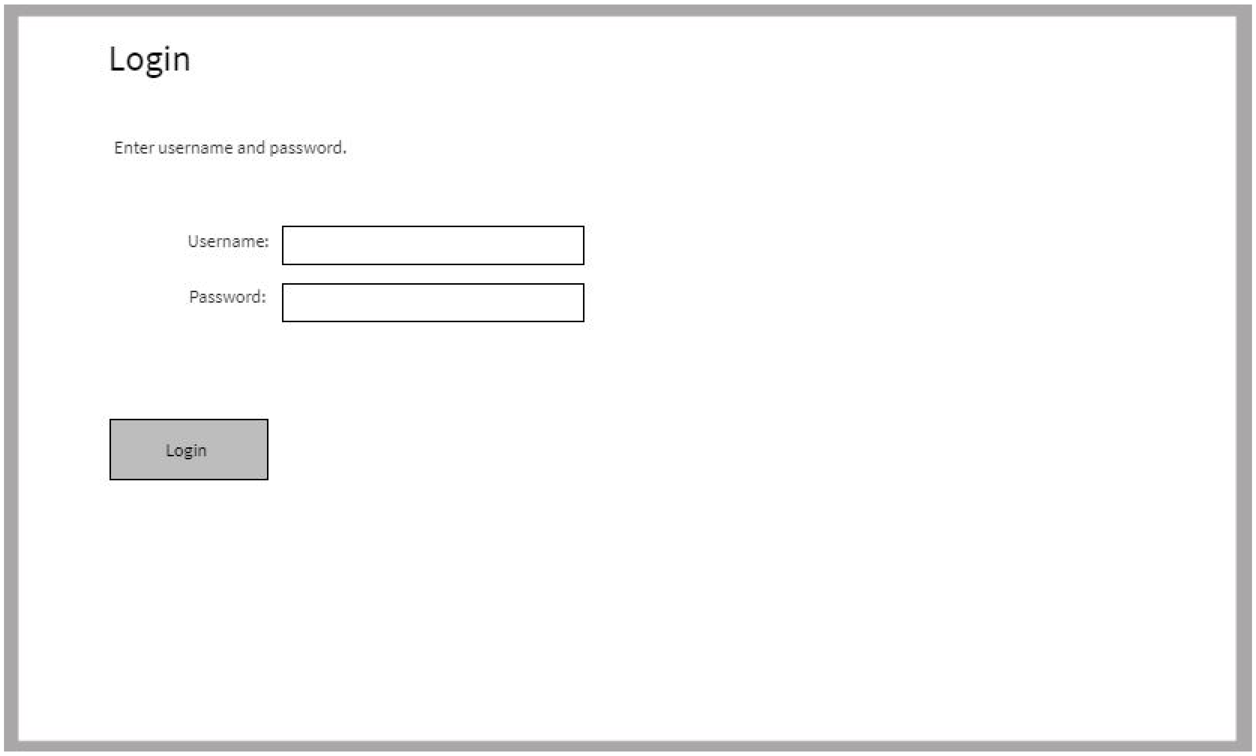
Home

Welcome to the Blockchain based certificate verification program you can check your certificate's authenticity or apply for verfication.

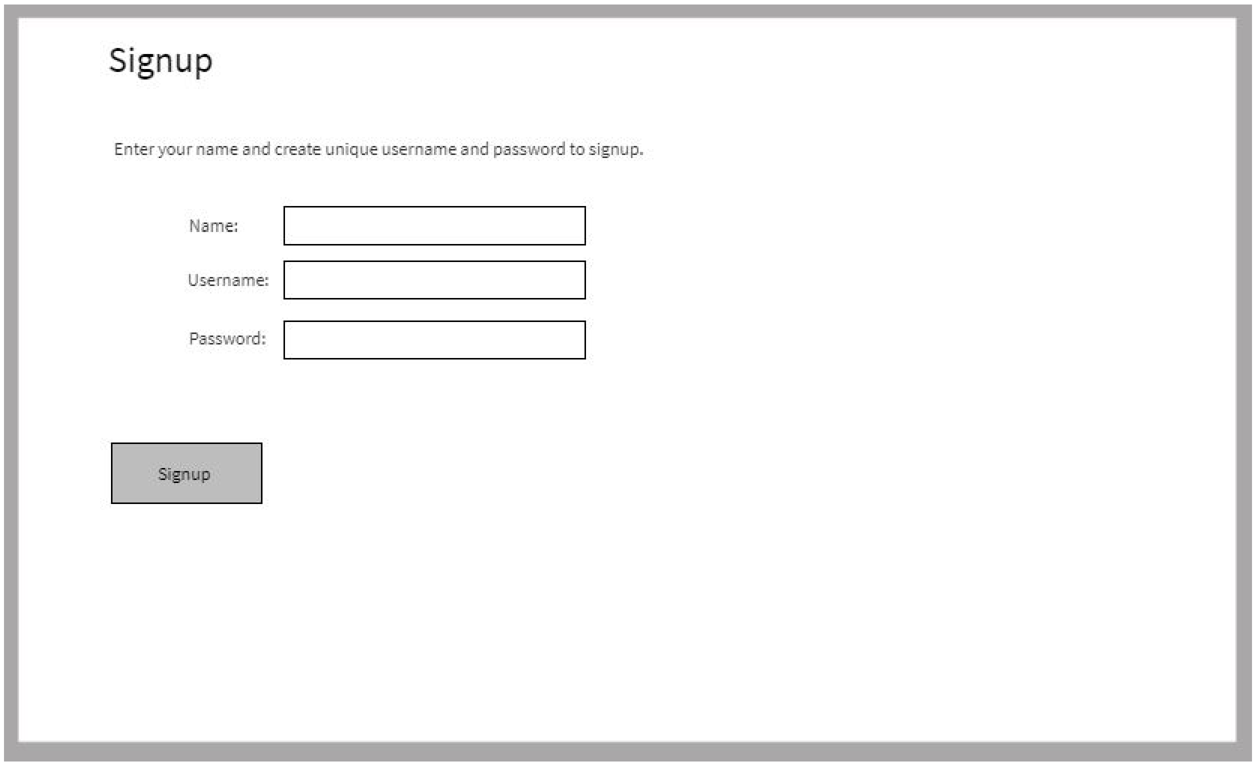
Login to apply for verification or click the signup button to create a new account.

Click the 'Check' button to check your certificate's authenticity .

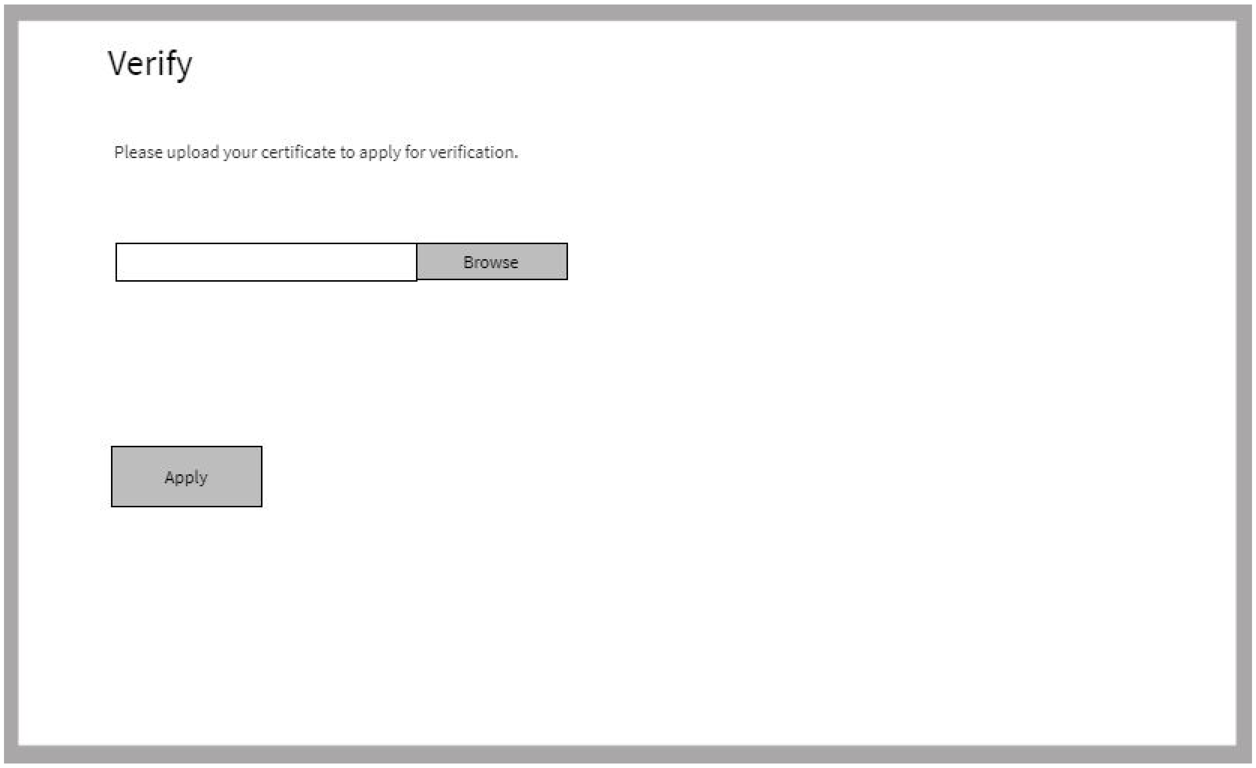
3 Figure 3 – 2



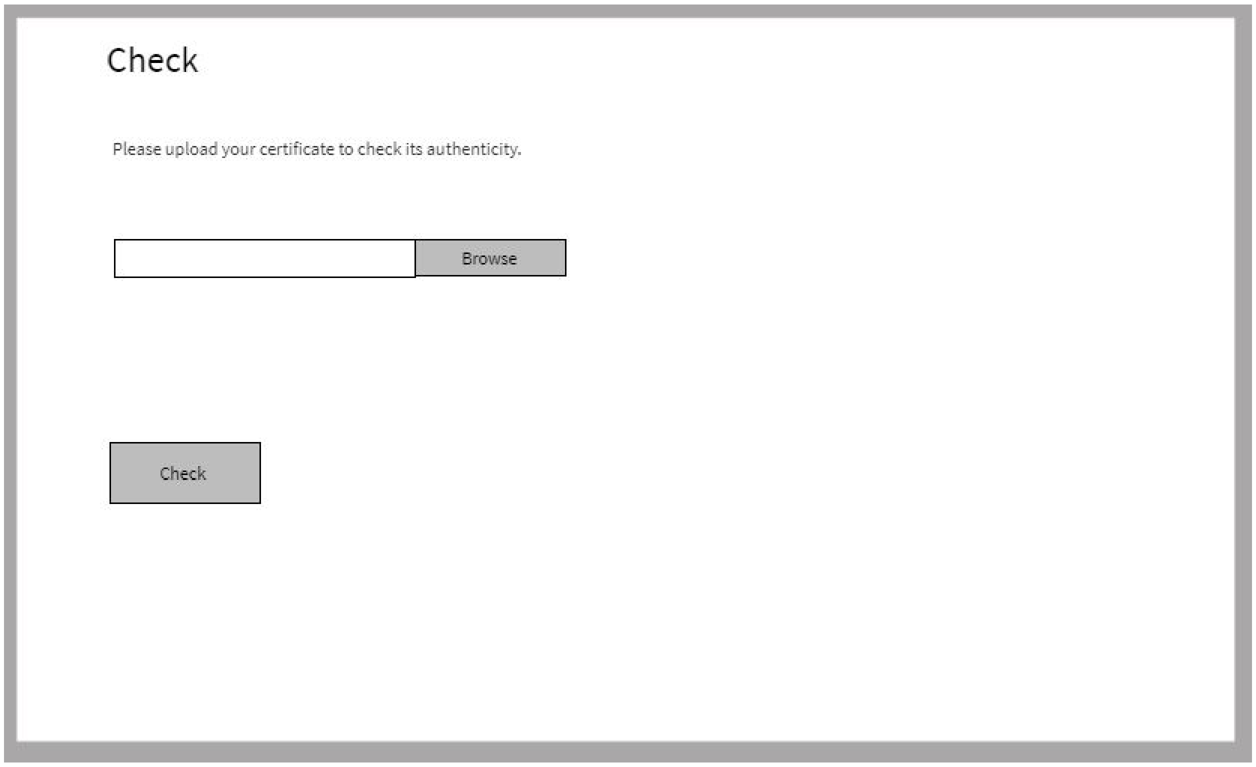
4 Figure 3 – 3


Signup
Enter your name and create unique username and password to signup.
Name:
Username:
Password:
Signup

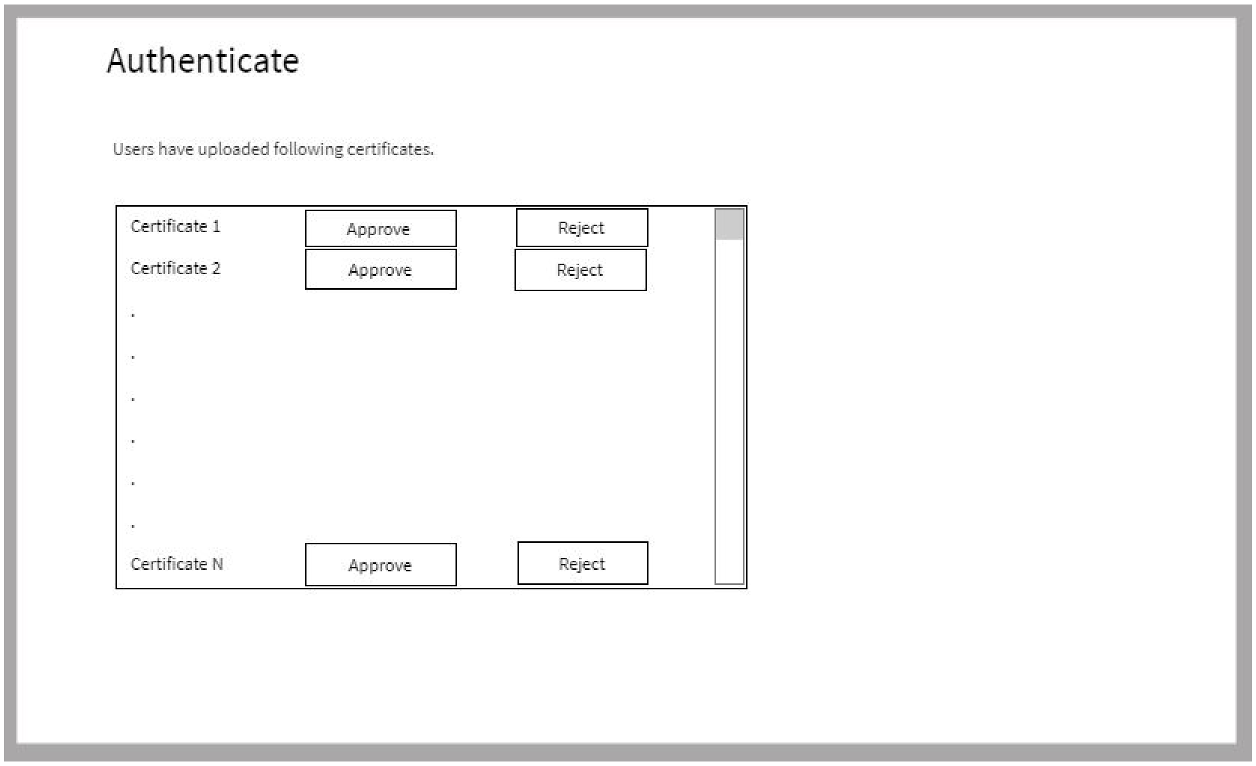
5 Figure 3 – 4



6 Figure 3 – 5



7 Figure 3 – 6



8 Figure 3 – 7

3.3.2 Hardware Interfaces

We do not require any hardware for this project.

3.3.3 Software Interfaces

This Degree Verification System will be developed on Linux, Ubuntu 16.10 or 18.04, and will use IDE VS Code, React App, Ganache, Metamask and R

JavaScript enabled web browser will be required to run this tool.

3.3.4 Communication Interfaces

Internet Connection will be needed.

3.4 Non-Functional Requirements

3.4.1 Performance Requirements

3.4.1.1 Reliability

Reliability is defined as providing the user up to date, correct information when they need it.

3.4.1.2 Security

System shall be secure enough not to breach any security.

3.4.1.3 Legal

System should follow customer privacy policy strictly.

3.4.2 Safety Requirements

Information shall be safely and securely transmitted between the blocks with no changes.

3.4.3 Software Quality Attributes

3.4.3.1 Usability

The graphical user interface of app is to be designed with usability as the first priority. The app will be presented and organized in a manner that is both visually appealing and easy to use for every individual.

3.4.3.2 Accuracy

To ensure reliability and correctness, there will be zero tolerance for errors in the algorithm that computes results.

3.4.3.3 Reliability

This software will not fail in any condition.

3.4.3.4 Availability

The application will always be available unless user close the browser or his personal computer.

3.4.3.5 Flexibility

The design and architecture of the application will be flexible enough for catering any new requirements, if any at some later stage or for the application enhancement.

3.4.3.6 Transparency

The system shall provide transparency by being fraud free.

4 Chapter 4: Design and Development

4.1 Introduction

This design document contains all functional requirements and displays their relationships with each other conceptually. This document also shows our planning towards implementation of our project. It contains diagrams which shows the design of the language and user interaction with the language followed by their responses. This document also consists of some tradeoffs of few aspects of the design, intended to be included.

4.1.1 Purpose of this Document

The software design document describes the architecture and the system design of the project Degree verification system using Blockchain. The goal of this document is to give the **development team** guidance on the architecture of the system to be developed and implemented through examples of software structure, software component, interfaces and data structures. The audience or users for this design document include the **project supervisor** and **UG project evaluation team.**

4.1.2 Scope of the Development Project

The proposed system, is an online web application that will provide a tamper proof method of verifying certificates. With the convenience of only an internet connection, the user can instantly make an account, login and upload certificates, getting them verified thus saving time and being cost effective.

The application will enable the targeted users to verify an institution certificate online without having to come down to the school. All that is required of the employer is to enter verification link verify.

The verification system will bring great benefits to the society and the economy. Online certificate verification system improves the speed and quality of services of certificate authentication, promotes the globalization of markets and cut down costs. The service will be available for students, employers, placement consultancies, Institutions (Colleges, Universities).

4.1.3 Definitions, acronyms, and abbreviations

Ethereum: Ethereum is a public, blockchain-based distributed computing platform and operating system. It is an open-source platform that lets developers build and deploy decentralized applications using smart contracts programmed in a language called Solidity.

InterPlanetary File System: **IPFS** is a peer-to-peer network for storing and sharing data in a distributed file system. It is interoperable with smart contracts and **blockchain** data, so it can add reliable, low-cost storage capacity to the ethereum ecosystem.

4.1.4 Overview of document

The document is divided into sections and is already listed in the table of contents and figures list. However, here is a brief description of all the sections.

4.1.4.1.1 UML Class diagram

UML Class diagram further manifests the description of low level components of the software, thus making the system adequately comprehensible.

4.1.4.2 User Interface Issues

This section ponders upon the main principles of the project's user interface. Not touching about the technical details, the section is described by an overall diagram. Moreover, UML Activity diagrams, UML Sequence diagrams, and UI Design diagrams also elaborate the User Interface issues in a more intelligible manner.

4.1.4.2.1 UML Activity diagrams

UML Activity Diagrams follow a workflow-based approach to describe the overall functioning of the project. They are a very good means to see how various steps are involved in major tasks inside our project using a flow chart pattern without getting into the technical details.

4.1.4.2.2 UML Sequence diagrams

UML Sequence diagrams show how different steps are involved in the completion of a functionality of the project. They have a unique format that allows the reader to see how many graphical objects are used in a sequence for the completion of a system requirement.

4.1.4.3 UI Design

Some snapshots of graphical user interfaces are shown in this section that prototype the way a user shall be interacting with the system.

4.1.4.4 Detailed description of components

This section contains detailed description of all the major components of the system in a structured pattern (table), comprising of 10 x rows. The pattern (table) maintains symmetry in the document structure; and therefore it is followed for each of the components. Each part/row of the table is identified by a label, explaining the purpose of each point. The description of each point vis-à-vis the component being discussed, ponders upon the detailed account of it in the system.

4.1.4.5 Reusability and relationships to other products

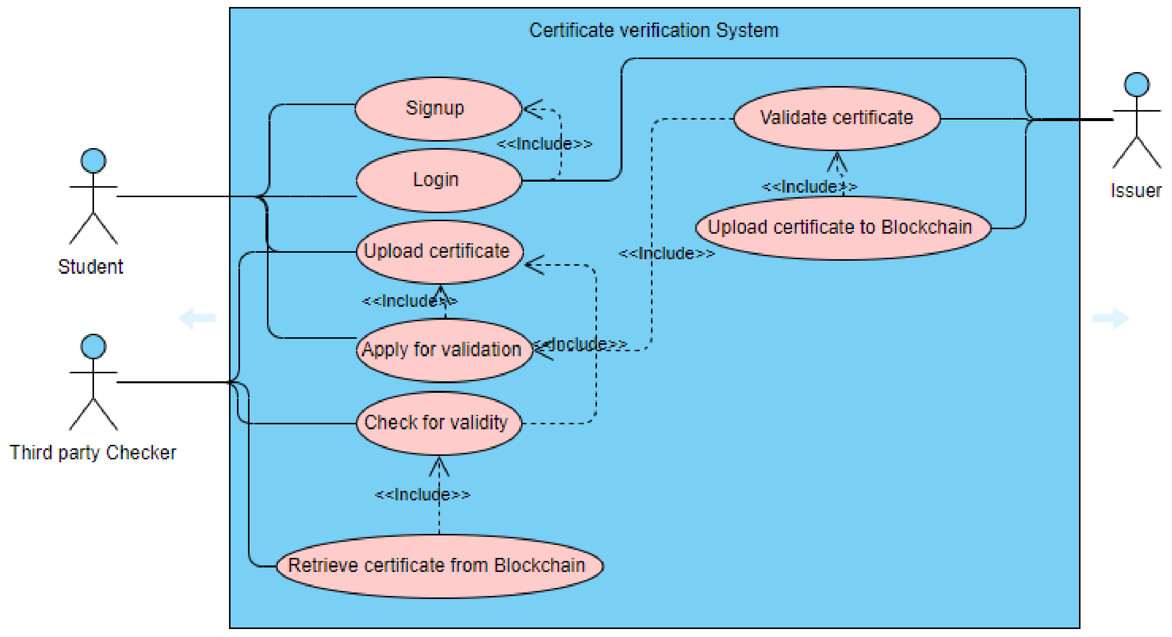
This section highlights the Reusability aspects of the various components of the system. Since the project in hand is all new and doesn’t carry out any enhancement work in the already existing system, so Reusability is just a recommended strategy to be employed while organizing various system components.

4.1.4.6 Design decisions and tradeoffs

This section focuses upon various design decisions and the ideas behind those. It enables the reader to understand the important crux of the design that is being used while excavating a bit more about the motivations behind those decisions.

4.2 System Architecture Description

4.2.1.1 Use Case Diagram



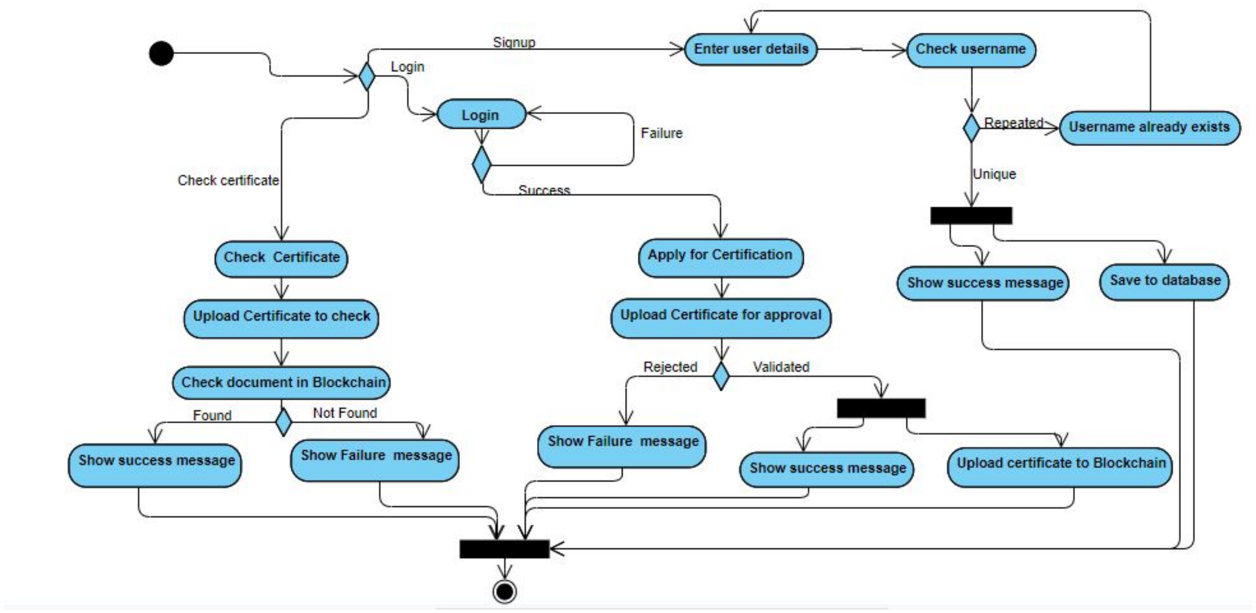
8 Figure 4 – 1 Use Case Diagram

4.2.1.2 UML Activity Diagram

This section shows the activities that a user need to perform to accomplish a task.

4.2.1.2.1 Student and Third Party Checker

Description: This scenario describes the flow of activities necessary for the verification of the degree/certificate.



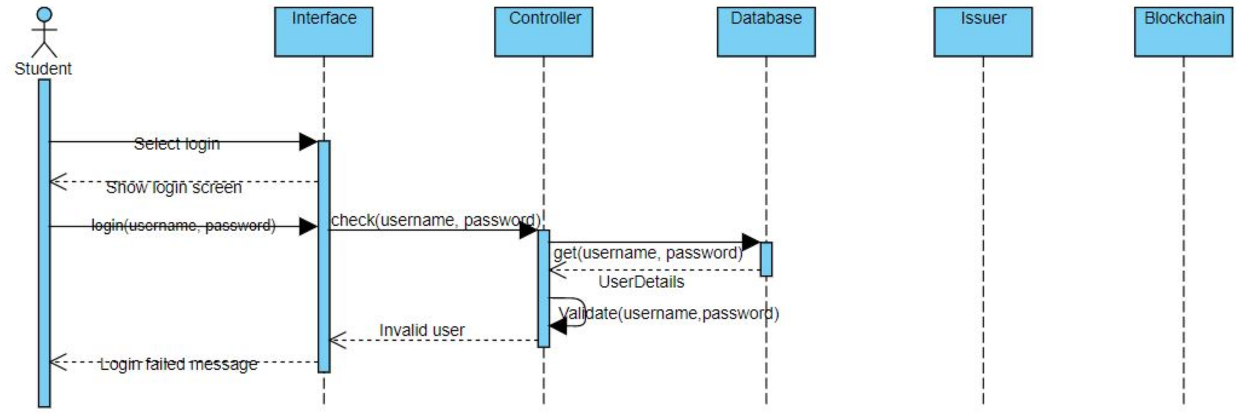
2Figure 4 - 7 Student and Third Party Checker

4.2.1.3 UML Sequence Diagrams

Different Scenarios and their corresponding events are discussed in this section with the help of sequence diagrams.

4.2.1.3.1 Student Login Failure

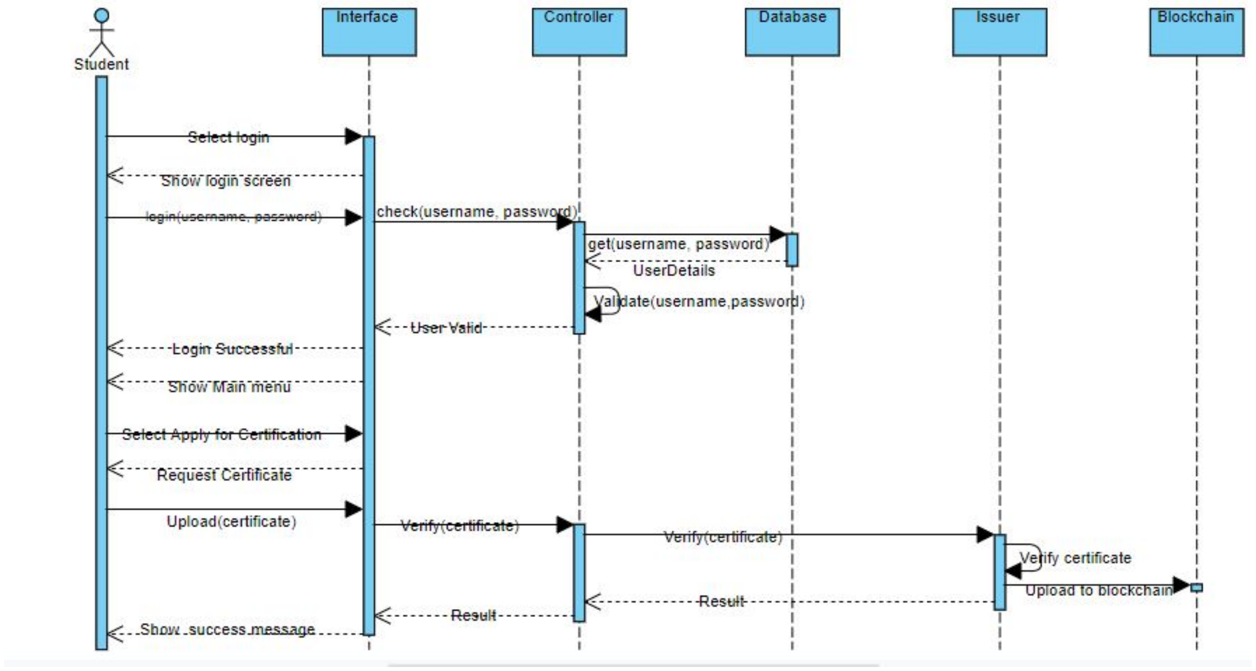
Description: This scenario describes the sequence of events that take place when there is a student login failure.



9 Figure 4 - 2 Student Login Failure Sequence

4.2.1.3.2 Successful Student Login and Certificate Verified

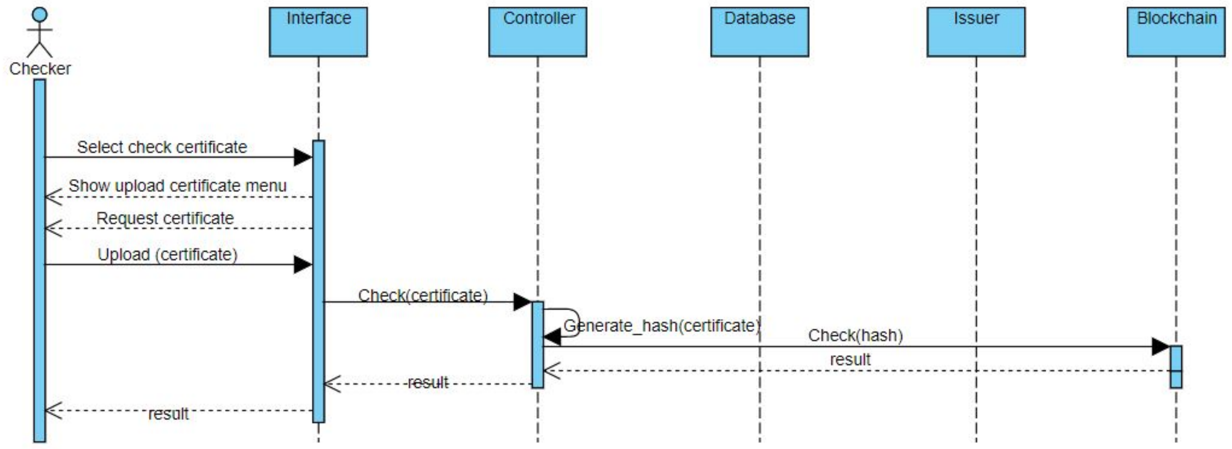
Description: This scenario describes the sequence of events that take place when there is successful student login and certificate gets verified.



10 Figure 4 - 3 Successful Student Login and Certificate Verified Sequence

4.2.1.3.3 Verify Certificate

Description: This scenario describes the sequence of events that take place when the certificate is verified.

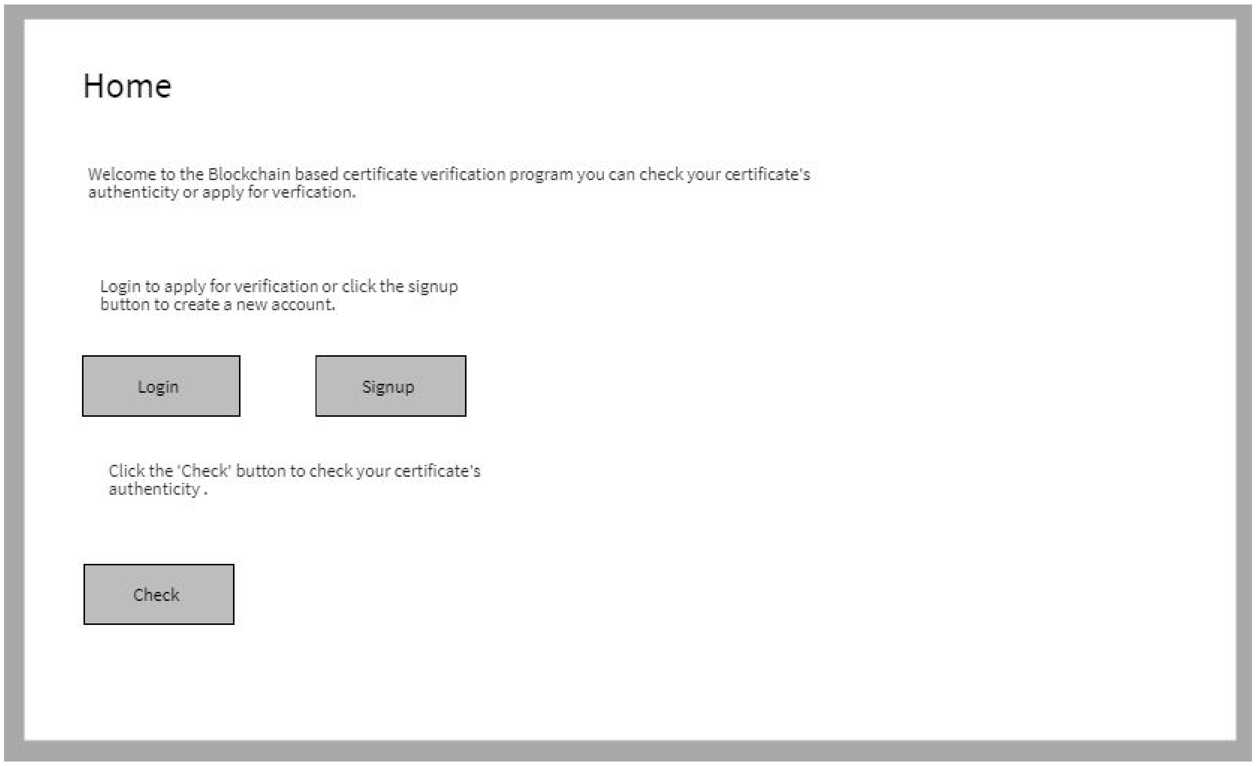


11 Figure 4 - 4 Verify Certificate Sequence

4.2.1.4 UI Design

The Degree Verification System is an online application and intended to be used by the users from diverse background knowledge. This requires that the interface of the system should have an easy learning curve for the user. Most of the important features should be visible to the user and no functionality should be hidden.

Please note that the interface provided is just for demonstration purposes. Actual interface may be different.



Home

Welcome to the Blockchain based certificate verification program you can check your certificate's authenticity or apply for verfication.

Login to apply for verification or click the signup button to create a new account.

Click the 'Check' button to check your certificate's authenticity .

12 Figure 4 - 5 Degree Verification System Interface

5 Chapter 5:Conclusion

The aim of this project was to help students, employers, placement consultancies, Colleges and universities in the process of verification of records and academic documents using the technological blockchain approach to prevent Forgery of Academic/Experience Certificates.

5.1 Purpose

The major targeted and achieved objectives are as follows:

- To ease verifying records and academic documents
- Allow everyone to check verified certificate
- To provide tamper proof method of storing certificates and providing verification.

5.2 Limitations

This analysis has focused only to decentralize the information of certificates. Using this project a Decentralized Document Management System can be designed on a larger scale with more information. Some limitations of this project are given below:

- Involvement of a centralized MySQL database
- Limited information of a certificate is stored inside Blockchain

5.3 Recommendations

To make this system more transparent and decentralized some recommendation as a future work are explained below:

Examining images of certificates along with the awardee details can also be stored as a permanent record inside theblock.

6 Bibliography

Ethereum, Ethereum Wallets
https://ethereum.org/en/

https://ethereum.org/en/wallets/

Ganache, Trufflesuite

Truffle (development framwork for dapps based on the Ethereum blockchain https://www.trufflesuite.com/)

Ganache (one click, in memory blockchain, https://www.trufflesuite.com/ganache)

Solidity Object-Oriented Programming Language

https://en.wikipedia.org/wiki/Solidity

Remix IDE for Solidity dApp (browser-based compiler)

https://remix.ethereum.org/

IPFS https://ipfs.io/

Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system."(2008).

Buterin, Vitalik. "A next-generation smart contract and decentralized application platform."White paper(2014).

Wood, Gavin. "Ethereum: A secure decentralisedgeneralised transaction ledger."Ethereum project yellow paper151 (2014):1-32.

Crosby, Michael, PradanPattanayak, Sanjeev Verma, and Vignesh Kalyanaraman. "Blockchain technology: Beyond bitcoin."Applied Innovation2, no. 6-10 (2016):71.

Swan M. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc."; 2015 Jan24.

https://www.academia.edu/40807150/A_Generic_Certificate_Verification_System_for_Nigerian_Universities_

Appendix

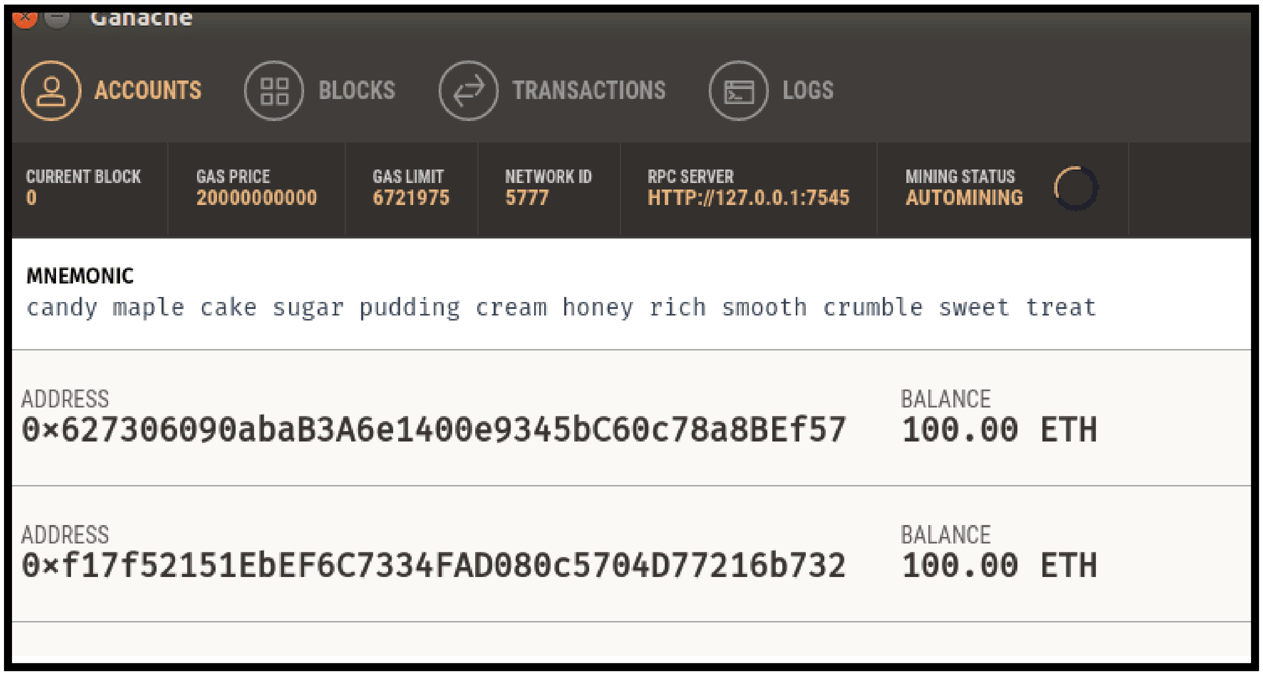
Tutorial for DApp to Store Certificates

The Application is designed to store Certificate images in an IPFS and generate hashes of the Certificates to store on the Ethereum blockchain. The steps taken to setting up the DApp are outlined below.

Step 1: Installing Ganache

We install Ganache while using the link https://github.com/trufflesuite/ganache/releases/tag/v1.2.2.Download its appropriate file for the Operating System and then follow installation instructions.

Run your Ganache, you will get a window screen that show you the account, the coins and other information.



CURRENT BLOCK 0 | GAS PRICE 20000000000 | GAS LIMIT 6721975 | NETWORK ID 5777 | RPC SERVER HTTP://127.0.0.1:7545 | MINING STATUS AUTOMINING

MNEMONIC
candy maple cake sugar pudding cream honey rich smooth crumble sweet treat

ADDRESS 0x627306090abaB3A6e1400e9345bC60c78a8BEf57 BALANCE 100.00 ETH

ADDRESS 0xf17f52151EbEF6C7334FAD080c5704D77216b732 BALANCE 100.00 ETH

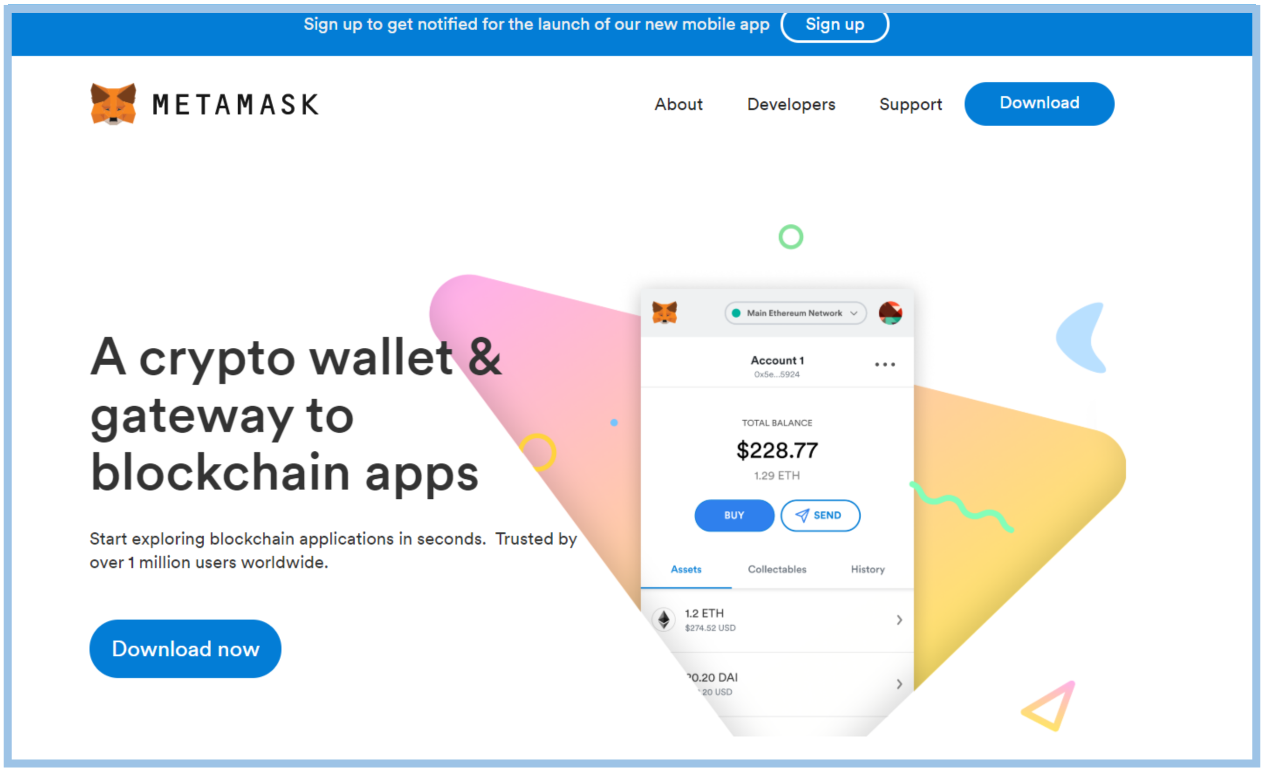
Step 2: Using Infura API for IPFS

To make the process more easy, the API provided by https://infura.io is used to access IPFS.

Step 3: Installing MetaMask

An Ethereum Wallet that allows us to access DApps in our browser. It provides user a secure interface to review the transactions the DApp wants to perform here.

Open https://metamask.io and get the Plugin for your browser.



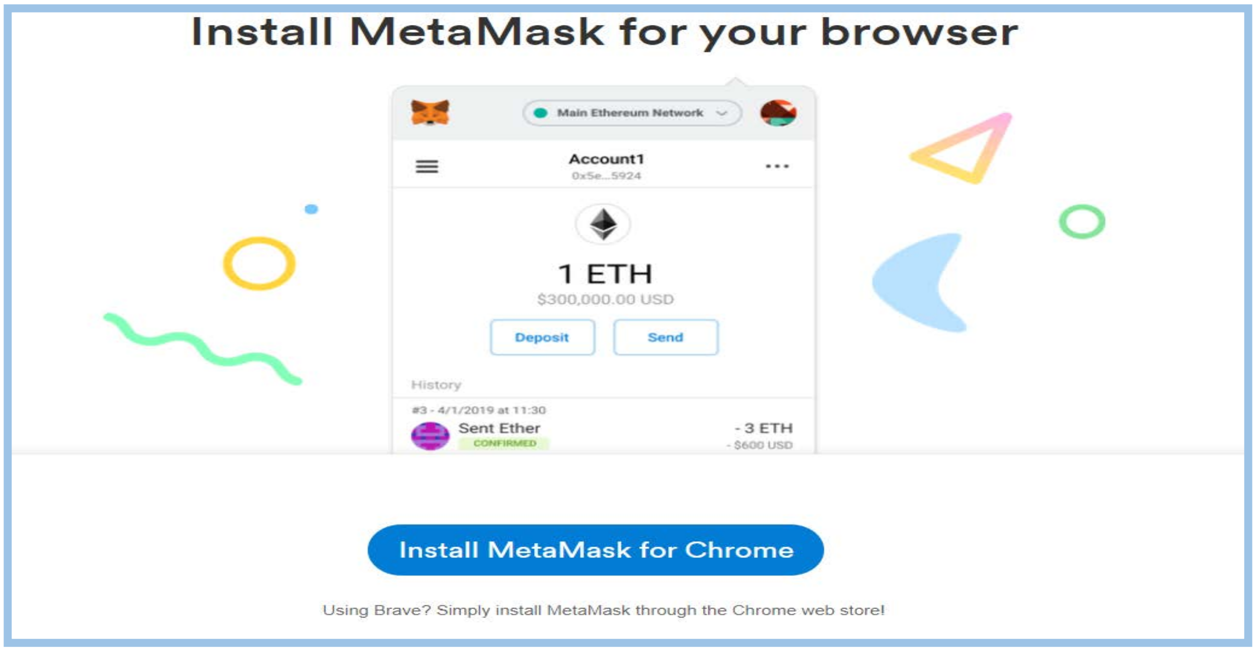
Sign up to get notified for the launch of our new mobile app Sign up

METAMASK About Developers Support Download

A crypto wallet & gateway to blockchain apps

Start exploring blockchain applications in seconds. Trusted by over 1 million users worldwide.

Download now

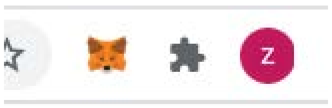


Install MetaMask for your browser

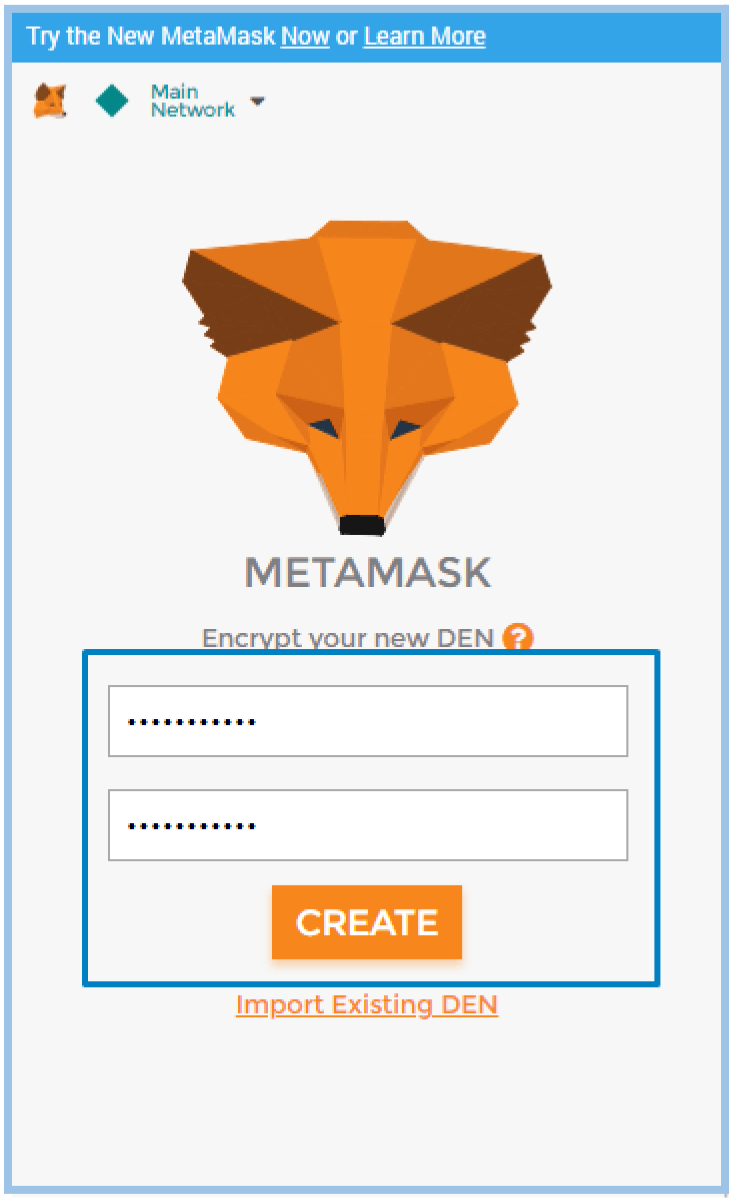
Install MetaMask for Chrome

Using Brave? Simply install MetaMask through the Chrome web store!

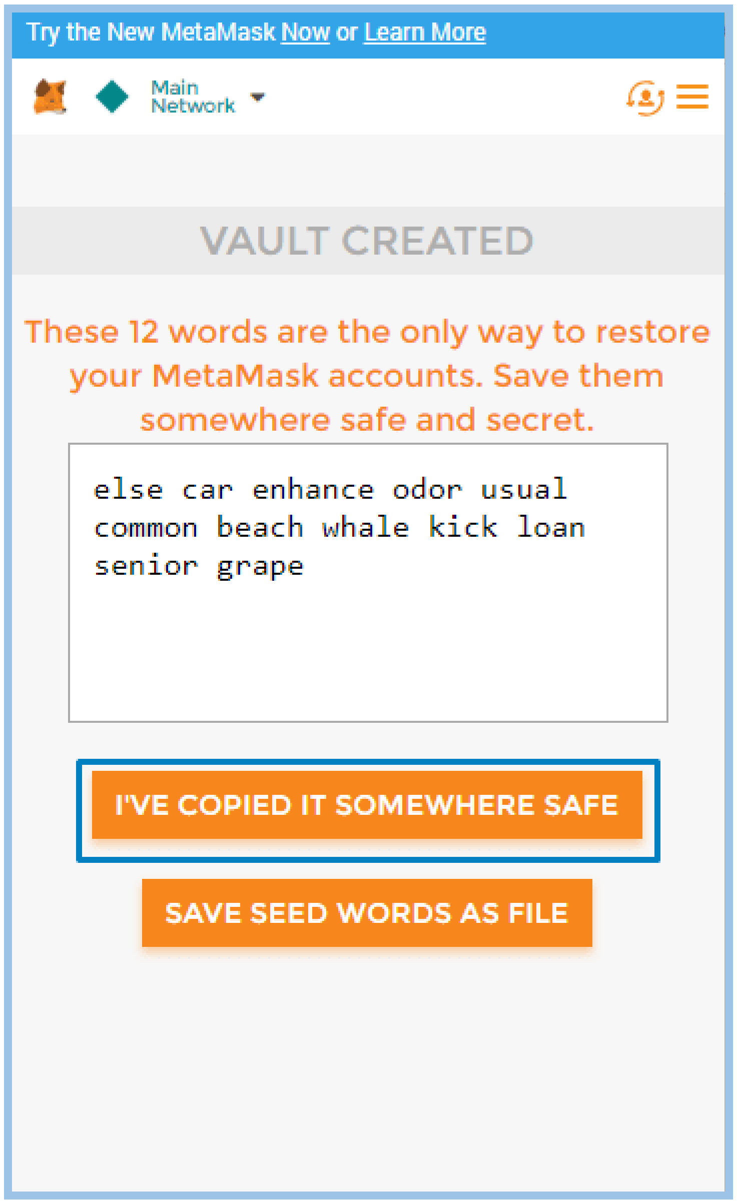
Now MetaMask should appear as an Extension in our chromium browser:



Now click on“Accept” to the next few windows until you are prompted to create a password**.**



After that screen contain keywords which is called Mnemonics. And these need to be stored somewhere in a secure place as they will allow us to access our wallet in case of a stolen or forgotten password. Now Click “I’ve copied it somewhere safe” as shown.



Try the New MetaMask Now or Learn More

Main Network

VAULT CREATED

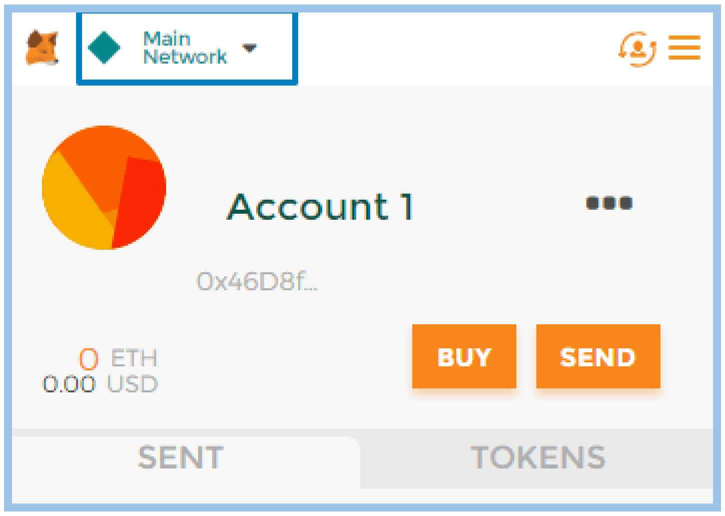
These 12 words are the only way to restore your MetaMask accounts. Save them somewhere safe and secret.

```
else car enhance odor usual
common beach whale kick loan
senior grape
```

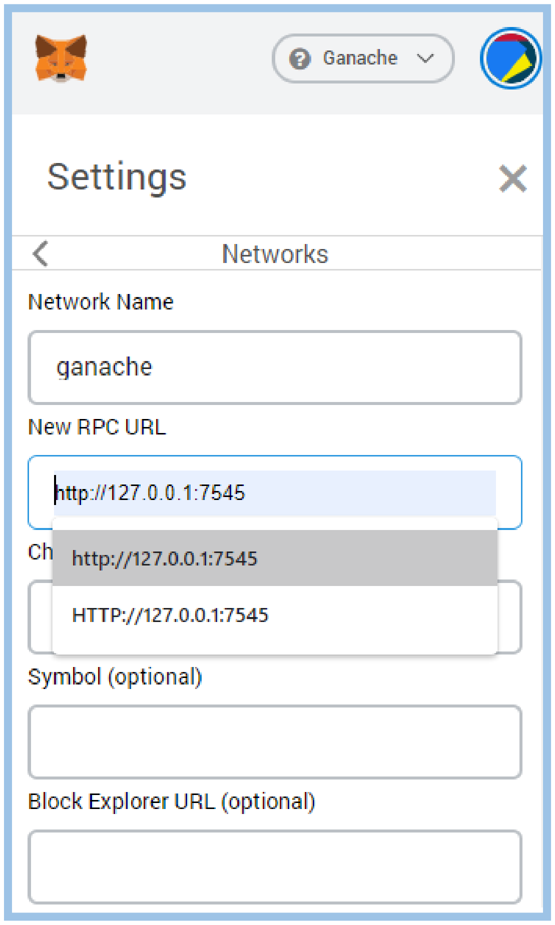
Step 4: Connect MetaMask wallet and Ganache.

For that go to top left corner of MetaMask and you will see “Main Network” with dropdown menu list. Click on arrow and click custom RPC where we will have to configure the network according to the RPC server details in the Ganache.

1.Click on this arrow, select custom RPC where we will have to configure the network aswith RPC server details of the Ganache.



2. These two fields are the most important to be filled.



Network Name

ganache

New RPC URL

http://127.0.0.1:7545

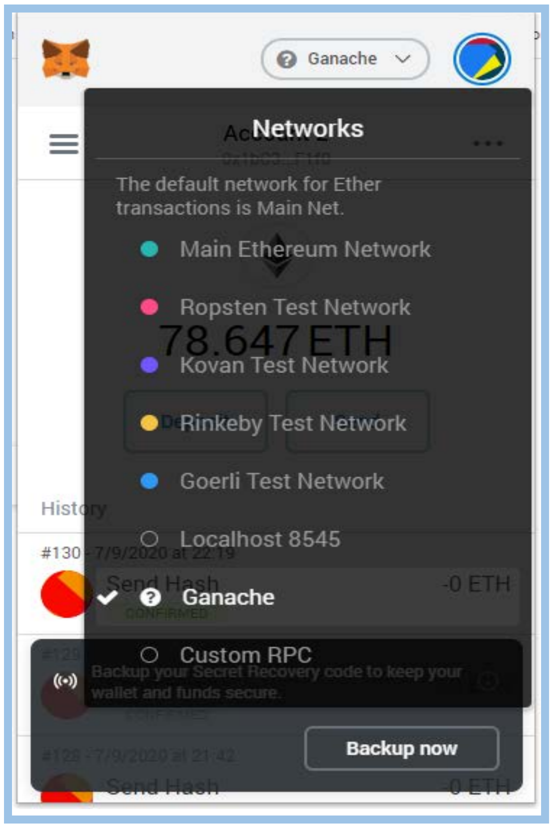
http://127.0.0.1:7545

HTTP://127.0.0.1:7545

Symbol (optional)

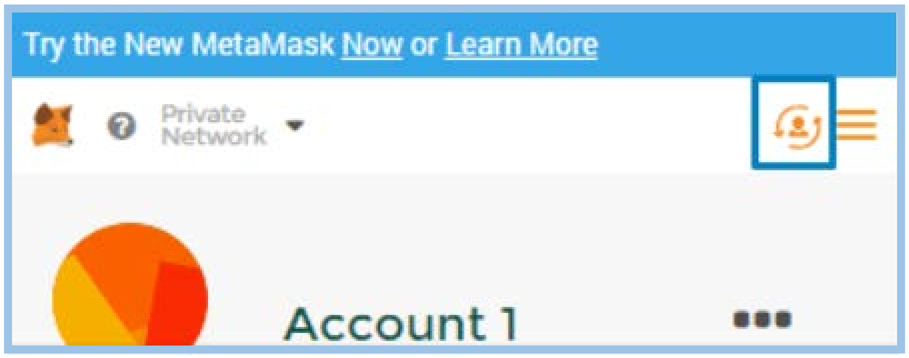
Block Explorer URL (optional)

3. With successful connection, our network shall be displayed.

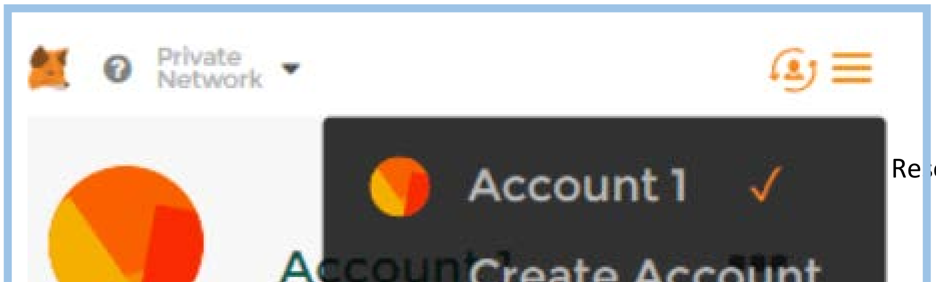


Step 5: Importing a Wallet

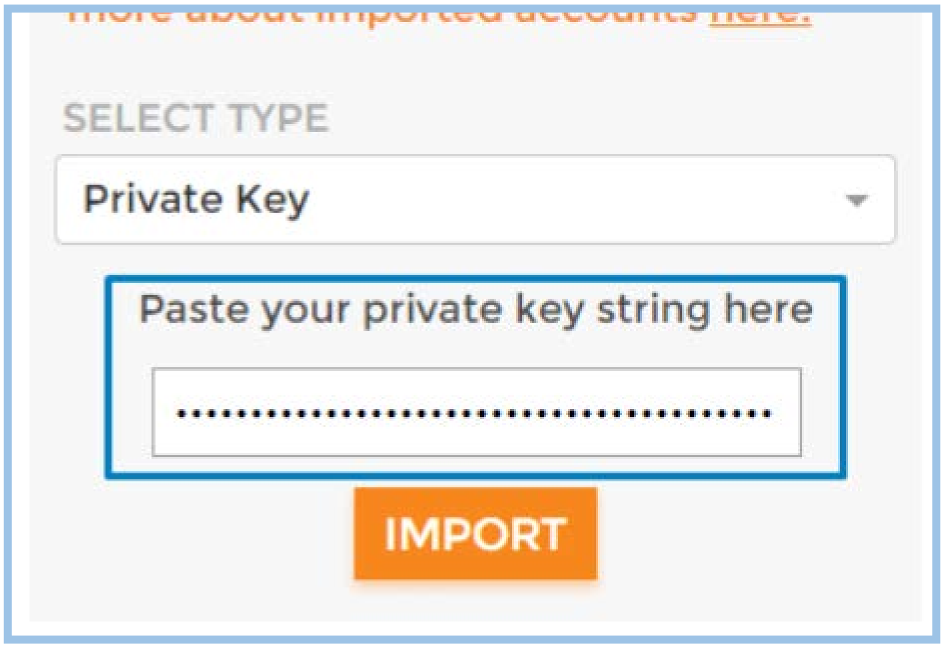
Open the Ganache and click on the Key icon with first account. After that copy Private Key to import it to our Wallet. Then open MetaMask and click on the shown icon:



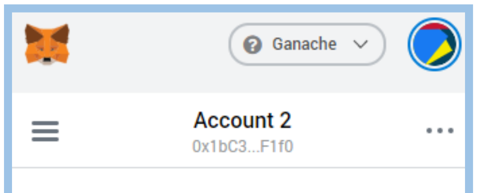
Select the "import account" option.



Now paste Private Key here into the box and click on “Import”.

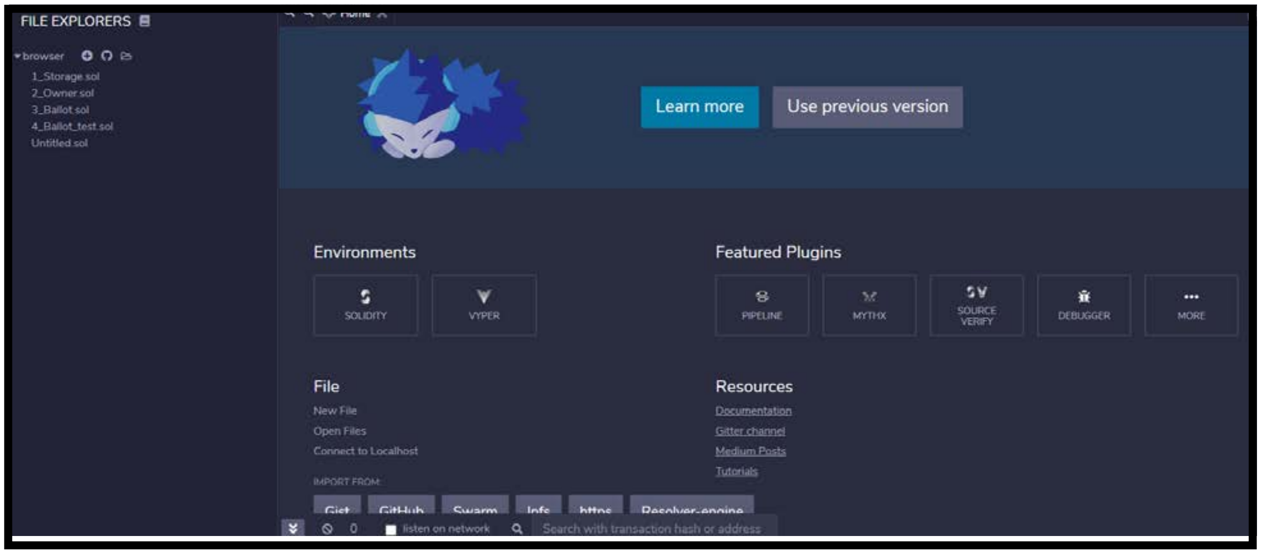


We now have some Ether in our account:

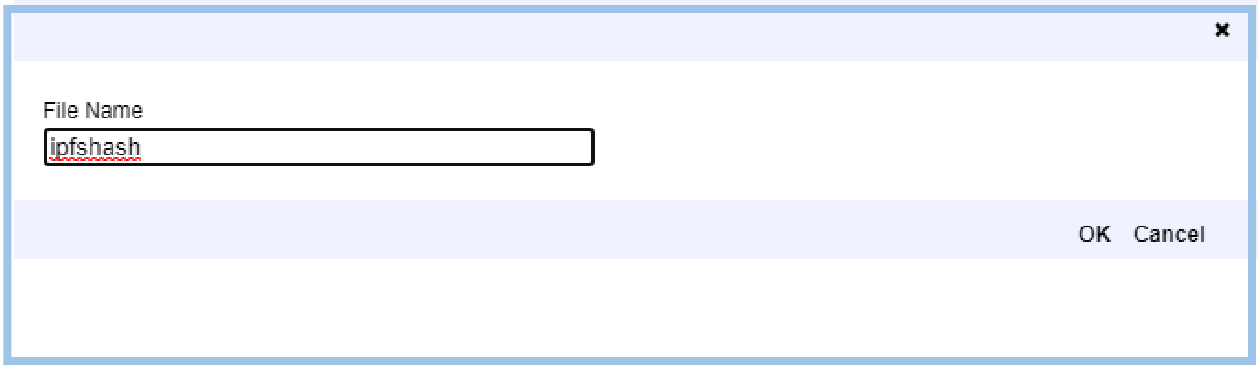


Step 6: Writing our Smart Contract

Now to write our smart contract, we going to use online editor called Remix. Open this link https://remix.ethereum.org/. Click on the plus in the left column in order to start our first contract.

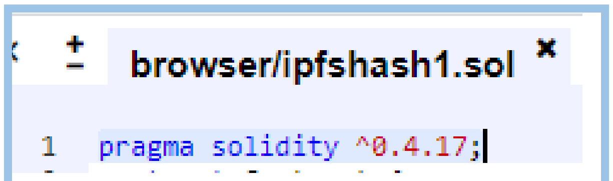


1: Now create file and give it a name.


File Name

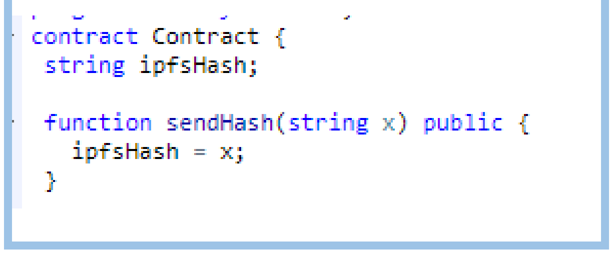
ipfshash

2. After creation of the file, start contract coding. We going to use language Solidity. The following indicate version of Solidity language we will be using:



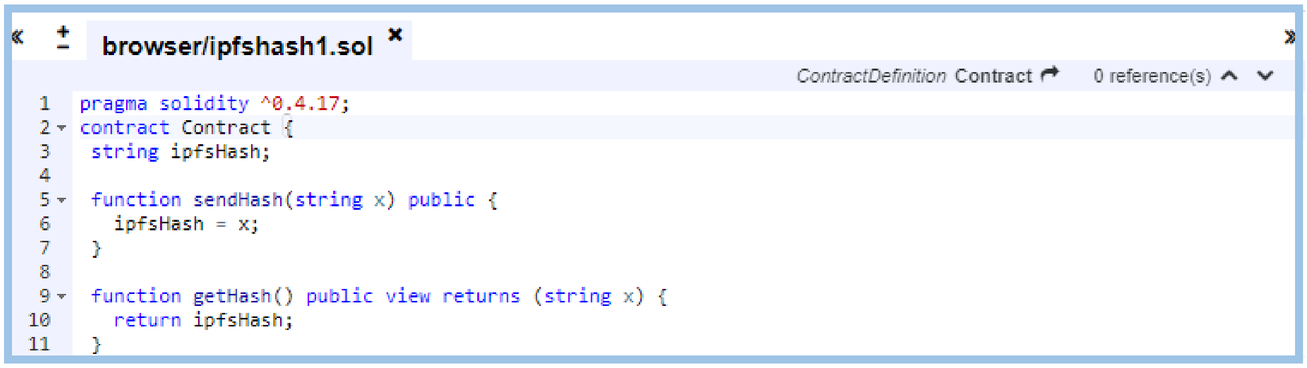
```
pragma solidity ^0.4.17;
```

3. Now define the contract:



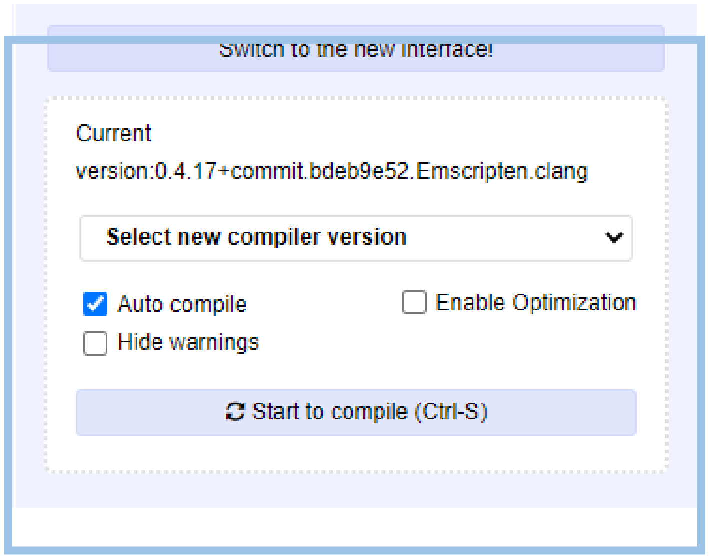
```
contract Contract {
 string ipfsHash;

 function sendHash(string x) public {
   ipfsHash = x;
 }
```

4. This smart contract here needed to store the hash of the uploaded file.



```
pragma solidity ^0.4.17;
contract Contract {
 string ipfsHash;

 function sendHash(string x) public {
   ipfsHash = x;
 }

 function getHash() public view returns (string x) {
   return ipfsHash;
 }
```

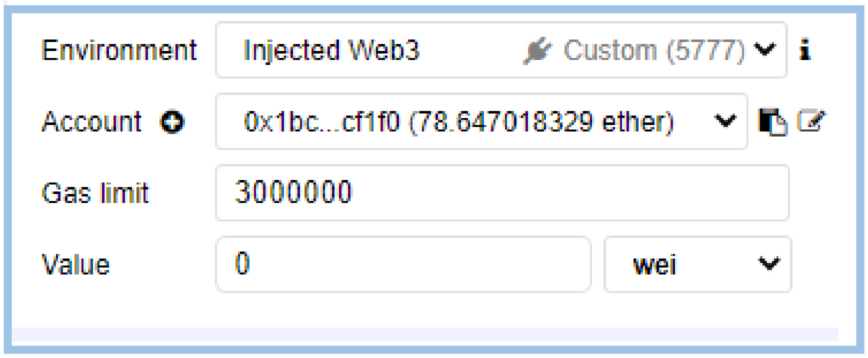
Now we should deploy the contract on our Ganache network. Now run the code.


Switch to the new interface!
Select new compiler version
Start to compile (Ctrl-S)

Current version:0.4.17+commit.bdeb9e52.Emscripten.clang

[✓] Auto compile
[ ] Enable Optimization
[ ] Hide warnings

Herebelow Environment option, click InjectedWeb3, Ganache network.



Environment: Injected Web3 — Custom (5777)

Account: 0x1bc...cf1f0 (78.647018329 ether)

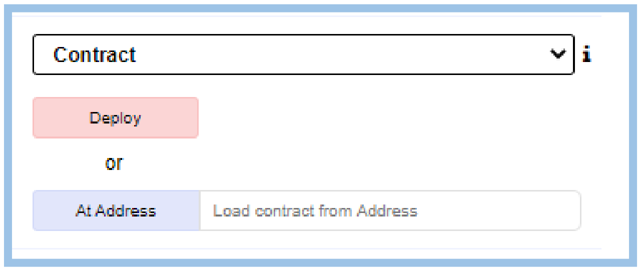
Gas limit: 3000000

Value: 0 wei

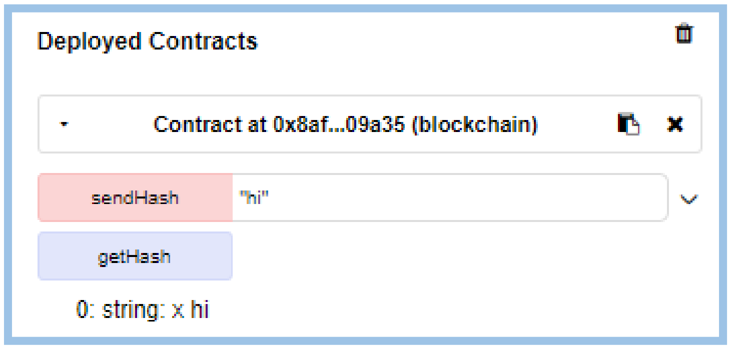
We will see a screen that asks if we are sure we want to connect to Ethereum node. Ok it.

Rimix going to fill the accounts from Ganache.

Now we will deploy contract by selecting the “Deploy” option.



Next, copy address of Contract from Deployed Contract section. We going to need this address later on to connect the contract and frontend.



Deployed Contracts

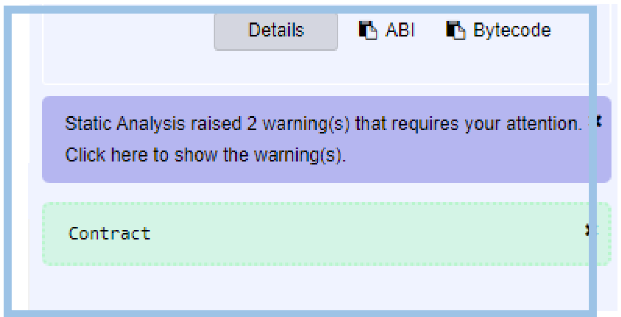
Contract at 0x8af...09a35 (blockchain)

sendHash "hi"

getHash

0: string: x hi

Now copy ABI of contract that will be required for the connection of frontend to the Contract.



Static Analysis raised 2 warning(s) that requires your attention.
Click here to show the warning(s).

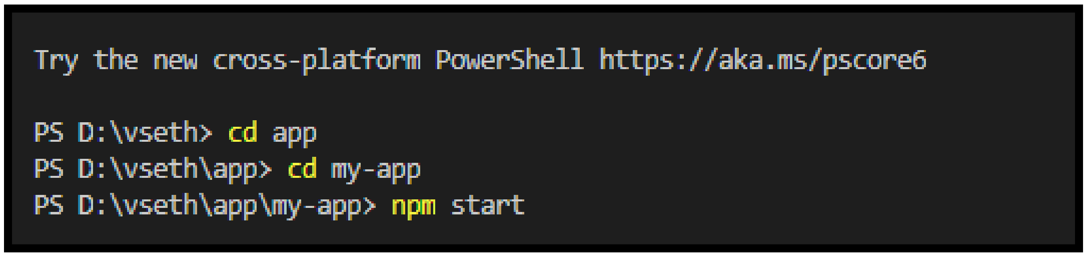
Contract

Step 7: Creating the JavaScript Frontend

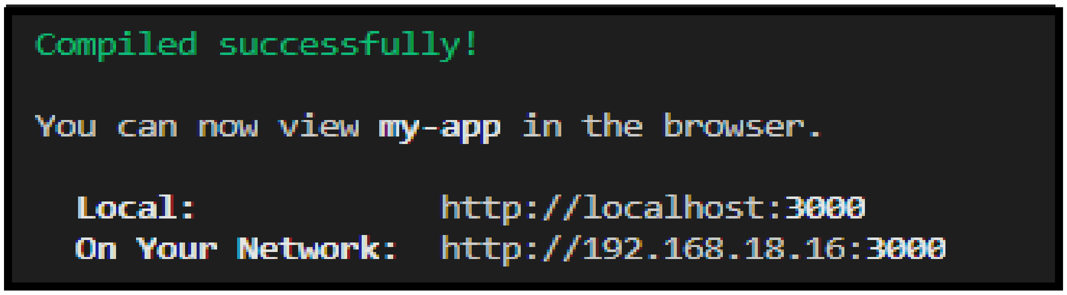
To upload the certificates.we need front end. So, we will use "React". **React an open-source JavaScript library** (also known as **React**. js or **ReactJS**) for building user interfaces.

So to work on REACT, we create a directory in which Dapp will be stored. And install following dependencies using this npm:
1.npm i create-react-app
2.npm install react-bootstrap
3.npm install fs-extra
4.npm install ipfs-api
5.npm install [web3@1.0.0-beta.26]()

Starting React-app:
1. Npm install –g create-react-app
2.Create-react-app my-app
3.cd into eth-ipfs and "npm start"

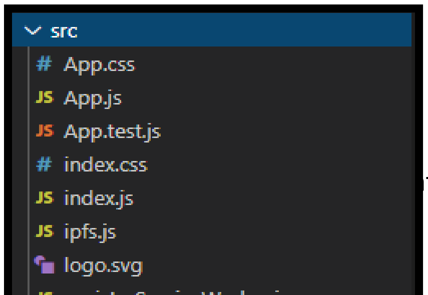


```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\vseth> cd app
PS D:\vseth\app> cd my-app
PS D:\vseth\app\my-app> npm start
```

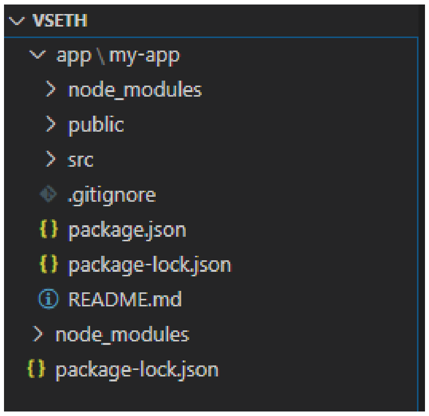


```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.18.16:3000
```

Structure of the directory:

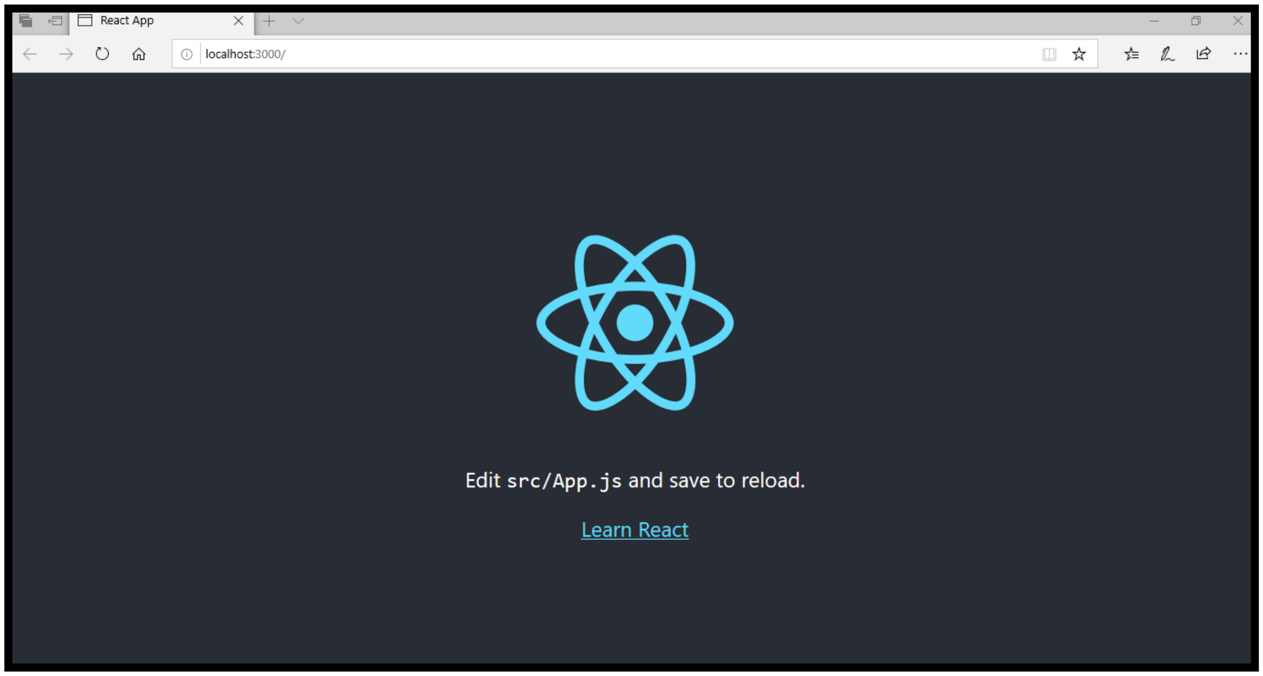


```
src
  # App.css
  JS App.js
  JS App.test.js
  # index.css
  JS index.js
  JS ipfs.js
  logo.svg
```



```
VSETH
  app \ my-app
    > node_modules
    > public
    > src
    .gitignore
    package.json
    package-lock.json
    README.md
  > node_modules
  package-lock.json
```

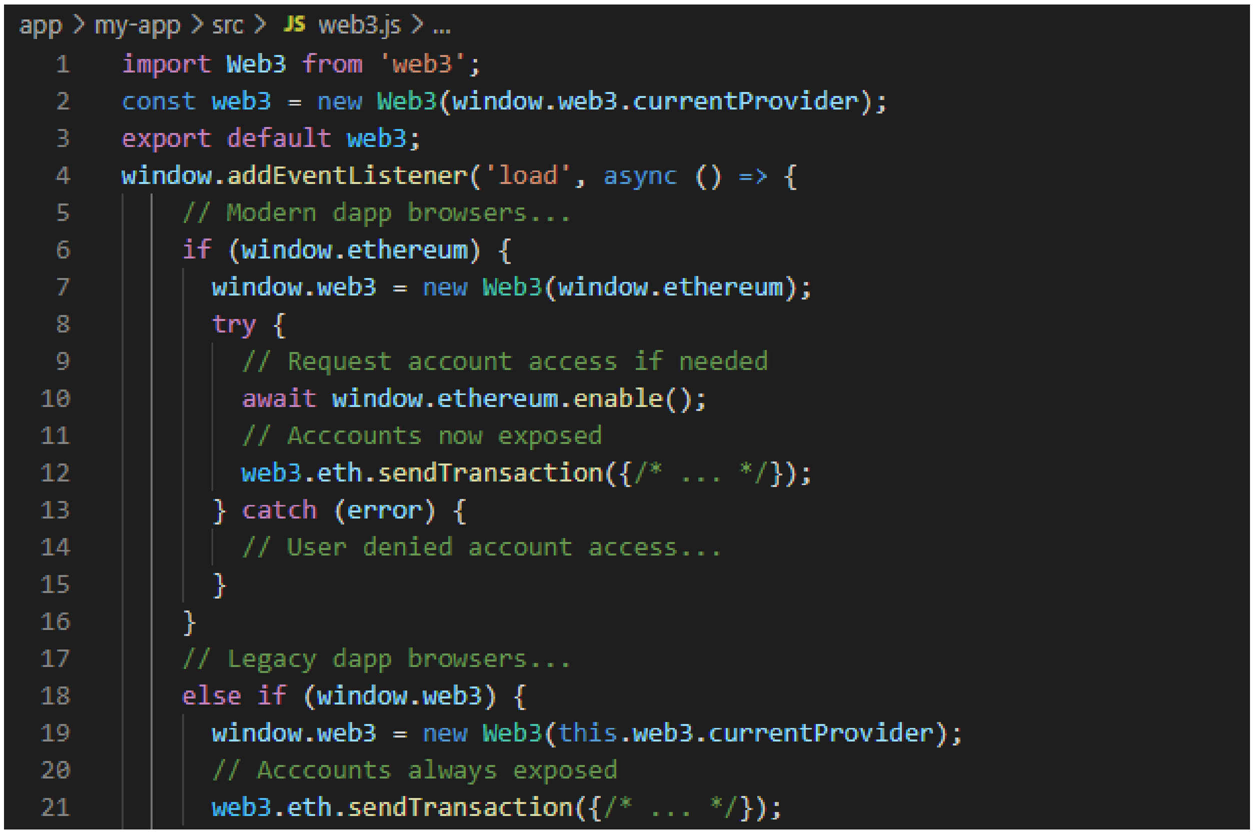
The below window shall be displayed on successful React app setup:



Edit `src/App.js` and save to reload.

Learn React

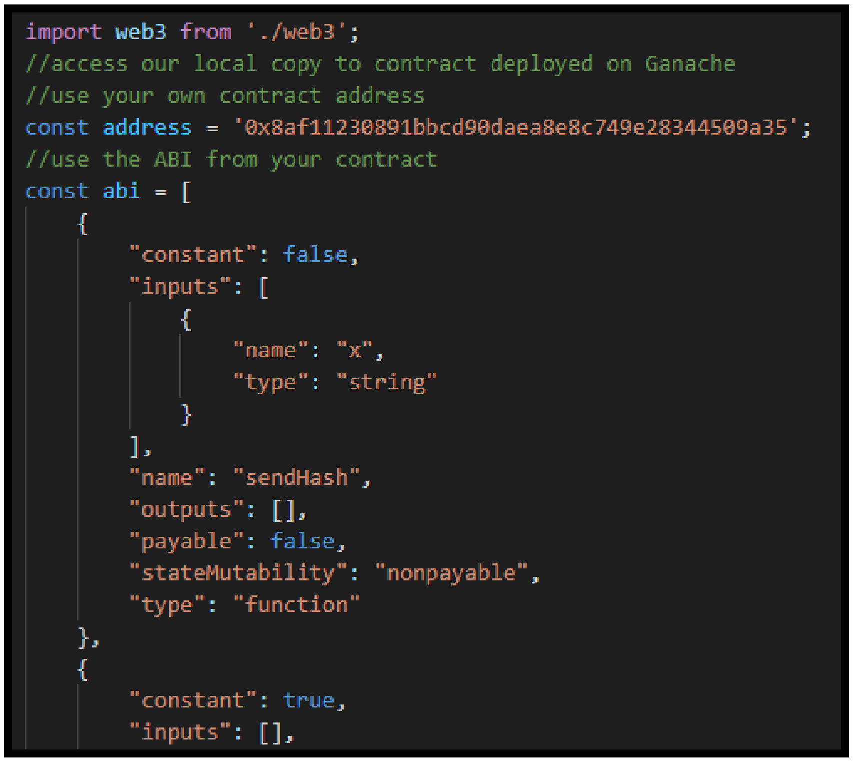
1. Now first need to create a file web3.js in “ src” which will be used by the Dapp, for recognition of metamask and connection with ethereum block chain:



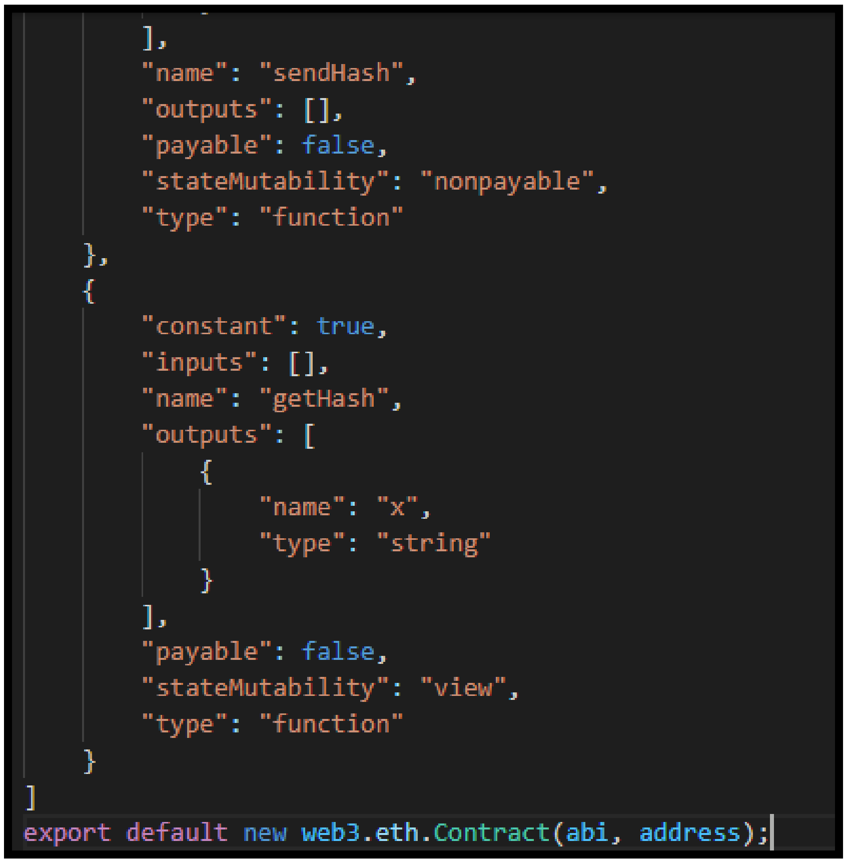
app > my-app > src > JS web3.js > ...

```
import Web3 from 'web3';
const web3 = new Web3(window.web3.currentProvider);
export default web3;
window.addEventListener('load', async () => {
    // Modern dapp browsers...
    if (window.ethereum) {
      window.web3 = new Web3(window.ethereum);
      try {
        // Request account access if needed
        await window.ethereum.enable();
        // Acccounts now exposed
        web3.eth.sendTransaction({/* ... */});
      } catch (error) {
        // User denied account access...
      }
    }
    // Legacy dapp browsers...
    else if (window.web3) {
      window.web3 = new Web3(this.web3.currentProvider);
      // Acccounts always exposed
      web3.eth.sendTransaction({/* ... */});
```

2.In next step, we will create a file called as storehash.js in "src" which will store the contract address and ABI of for interaction with smart contract.



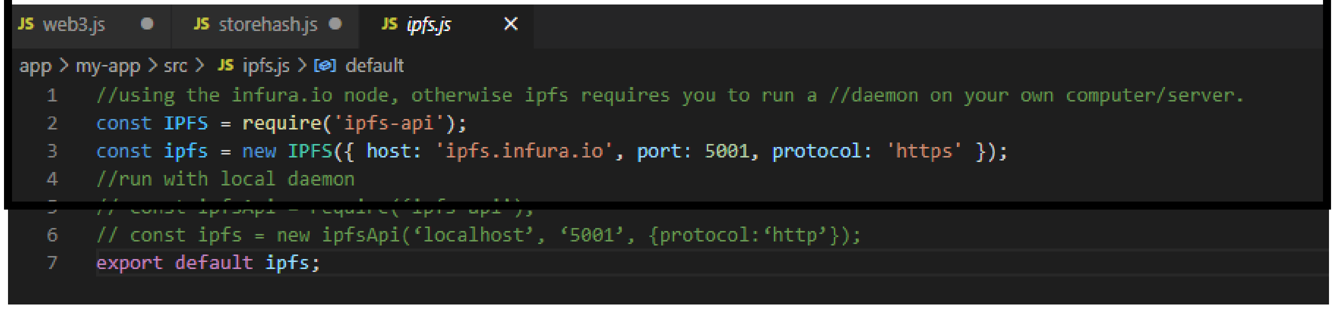
```
import web3 from './web3';
//access our local copy to contract deployed on Ganache
//use your own contract address
const address = '0x8af11230891bbcd90daea8e8c749e28344509a35';
//use the ABI from your contract
const abi = [
    {
        "constant": false,
        "inputs": [
            {
                "name": "x",
                "type": "string"
            }
        ],
        "name": "sendHash",
        "outputs": [],
        "payable": false,
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "constant": true,
        "inputs": [],
```



```
        ],
        "name": "sendHash",
        "outputs": [],
        "payable": false,
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "constant": true,
        "inputs": [],
        "name": "getHash",
        "outputs": [
            {
                "name": "x",
                "type": "string"
            }
        ],
        "payable": false,
        "stateMutability": "view",
        "type": "function"
    }
]
export default new web3.eth.Contract(abi, address);
```

3.

Another "ipfs.js" file will be created in "src" directory for connection with ipfs provided by infura.

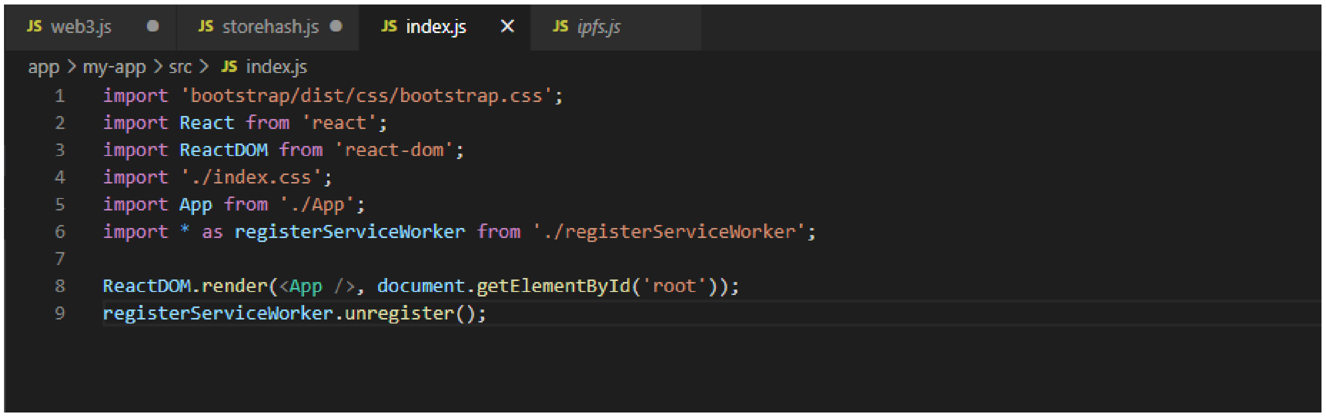


web3.js ● storehash.js ● ipfs.js ×

app > my-app > src > ipfs.js > default

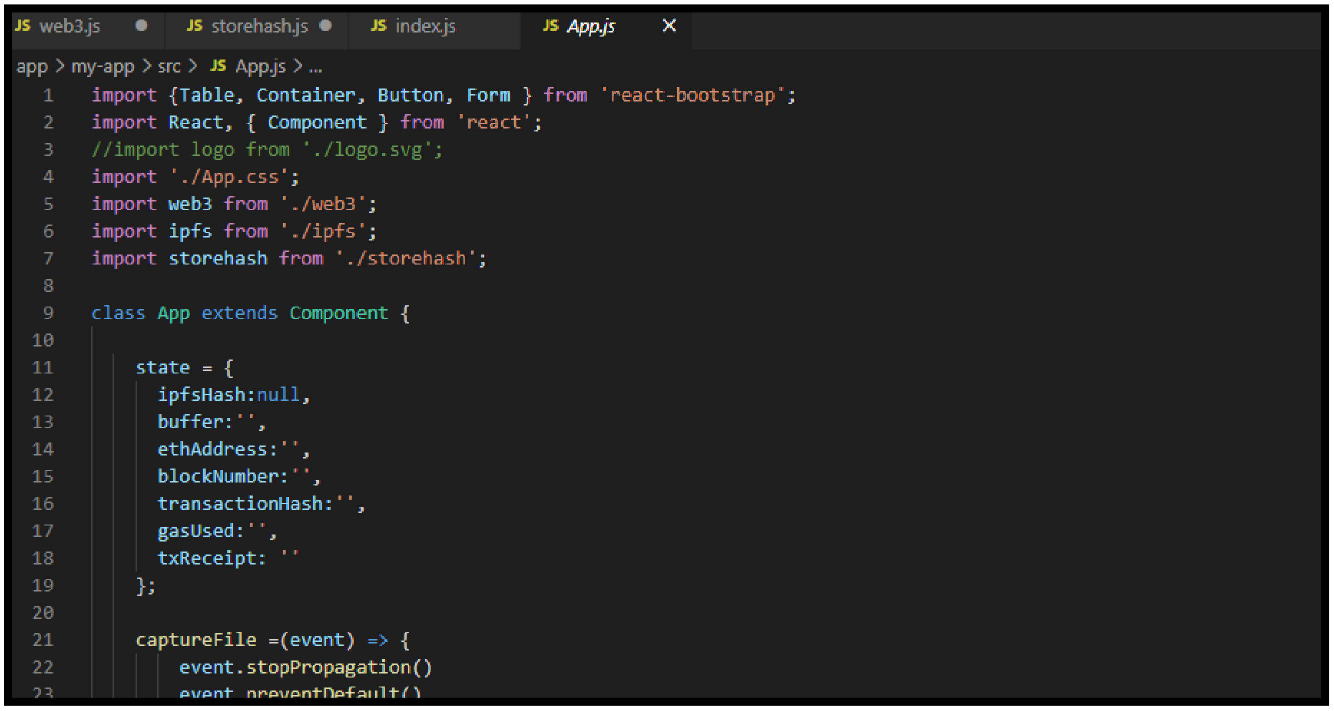
```
//using the infura.io node, otherwise ipfs requires you to run a //daemon on your own computer/server.
const IPFS = require('ipfs-api');
const ipfs = new IPFS({ host: 'ipfs.infura.io', port: 5001, protocol: 'https' });
//run with local daemon
// const ipfsApi = require('ipfs-api');
// const ipfs = new ipfsApi('localhost', '5001', {protocol:'http'});
export default ipfs;
```

4. Next step, edit the index.js file provided by the react app, when it's installed:

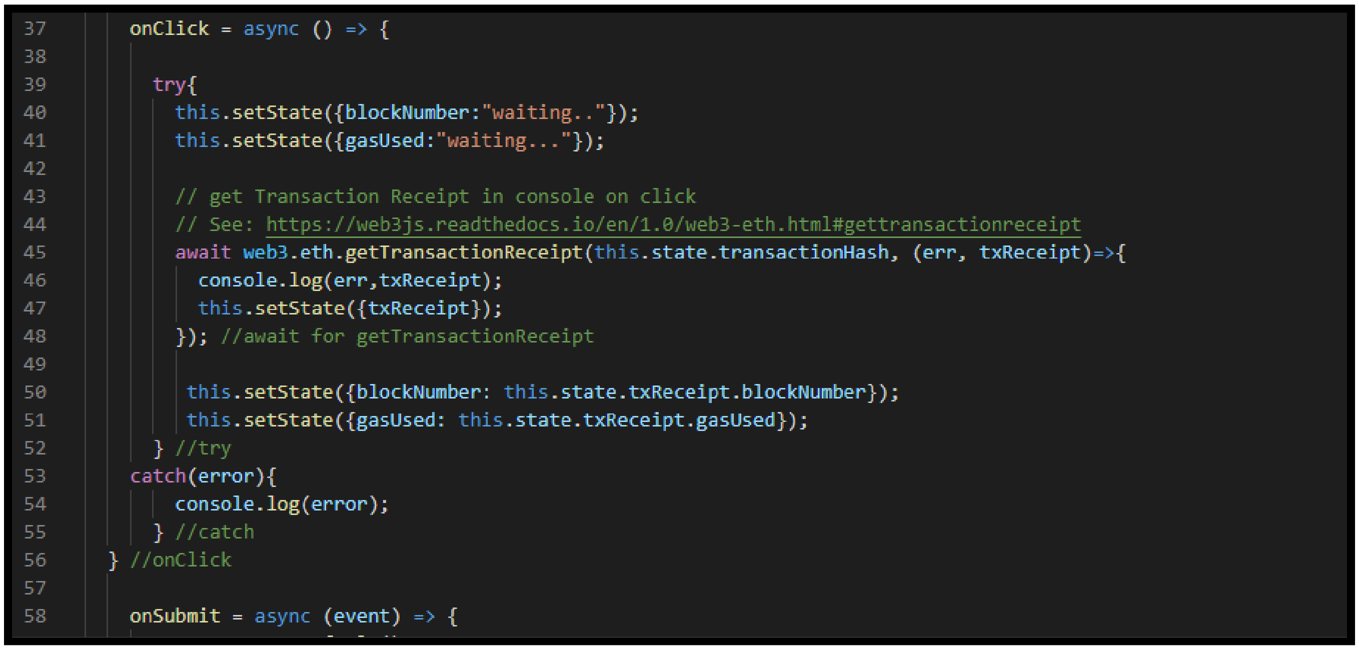


```
import 'bootstrap/dist/css/bootstrap.css';
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as registerServiceWorker from './registerServiceWorker';

ReactDOM.render(<App />, document.getElementById('root'));
registerServiceWorker.unregister();
```

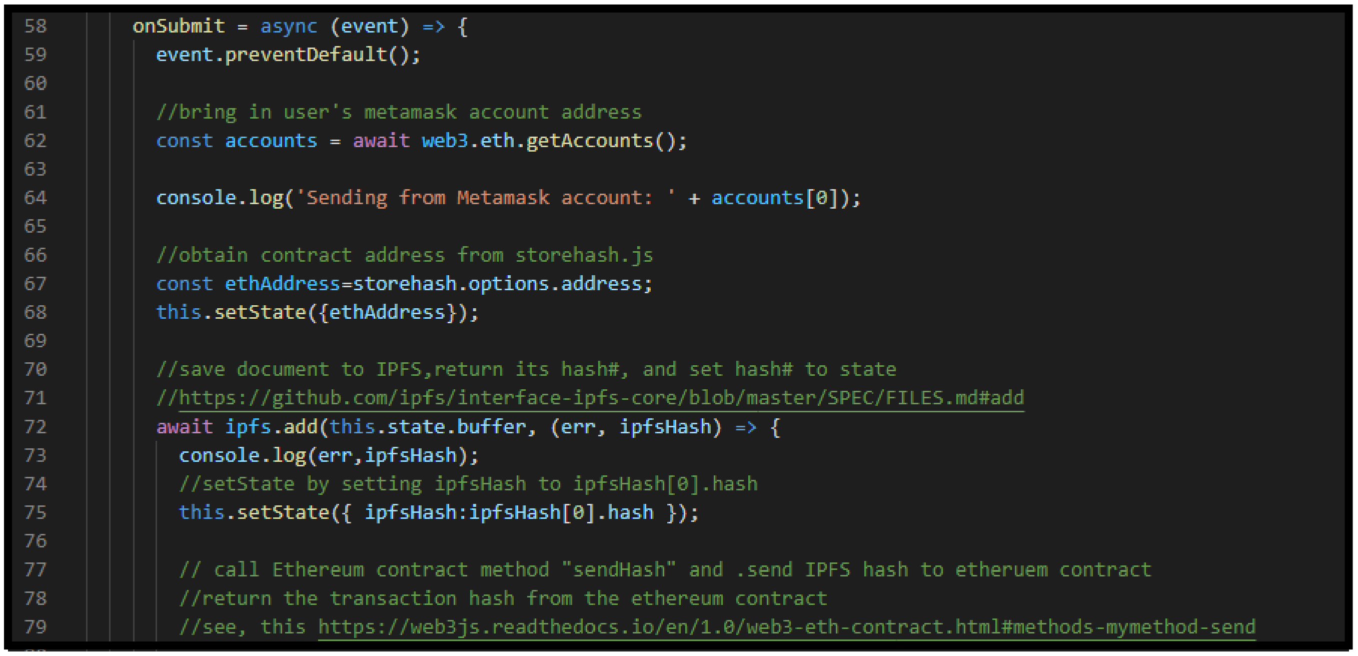
5.Last step, we will edit app.js file in src directory in order to create our Dapp:



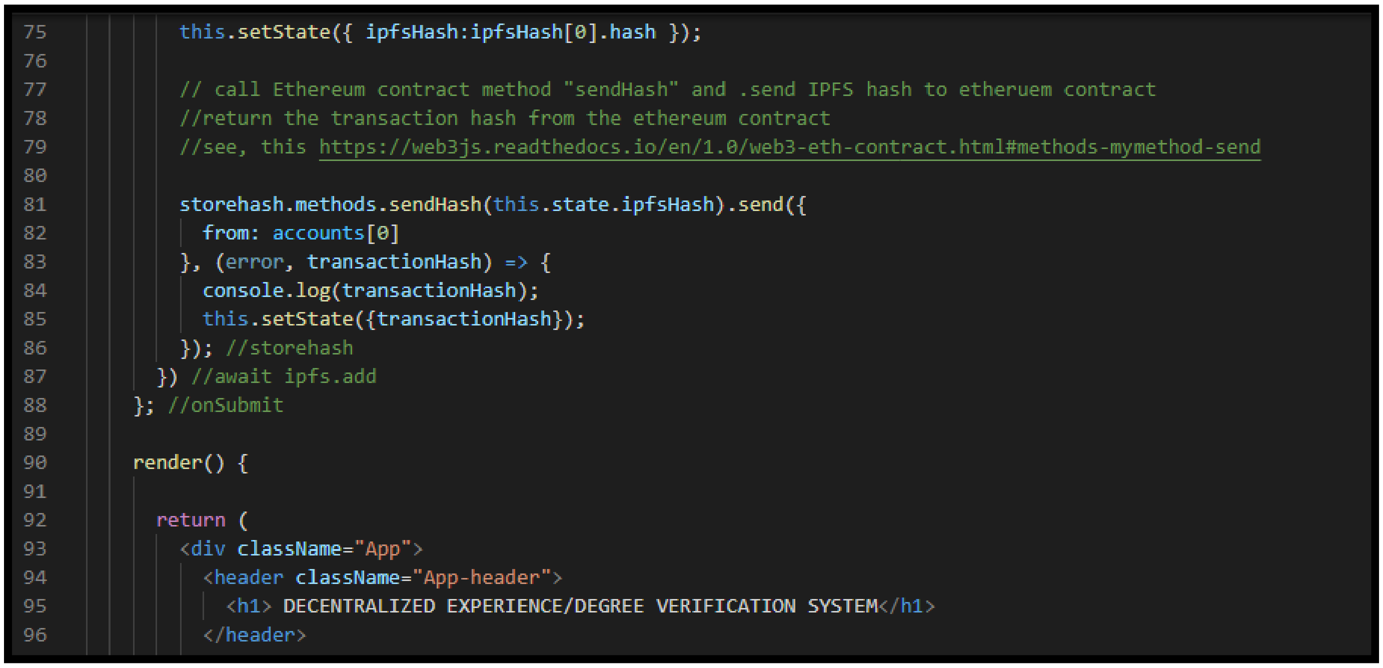
```
import {Table, Container, Button, Form } from 'react-bootstrap';
import React, { Component } from 'react';
//import logo from './logo.svg';
import './App.css';
import web3 from './web3';
import ipfs from './ipfs';
import storehash from './storehash';

class App extends Component {

    state = {
      ipfsHash:null,
      buffer:'',
      ethAddress:'',
      blockNumber:'',
      transactionHash:'',
      gasUsed:'',
      txReceipt: ''
    };

    captureFile =(event) => {
        event.stopPropagation()
        event.preventDefault()
```



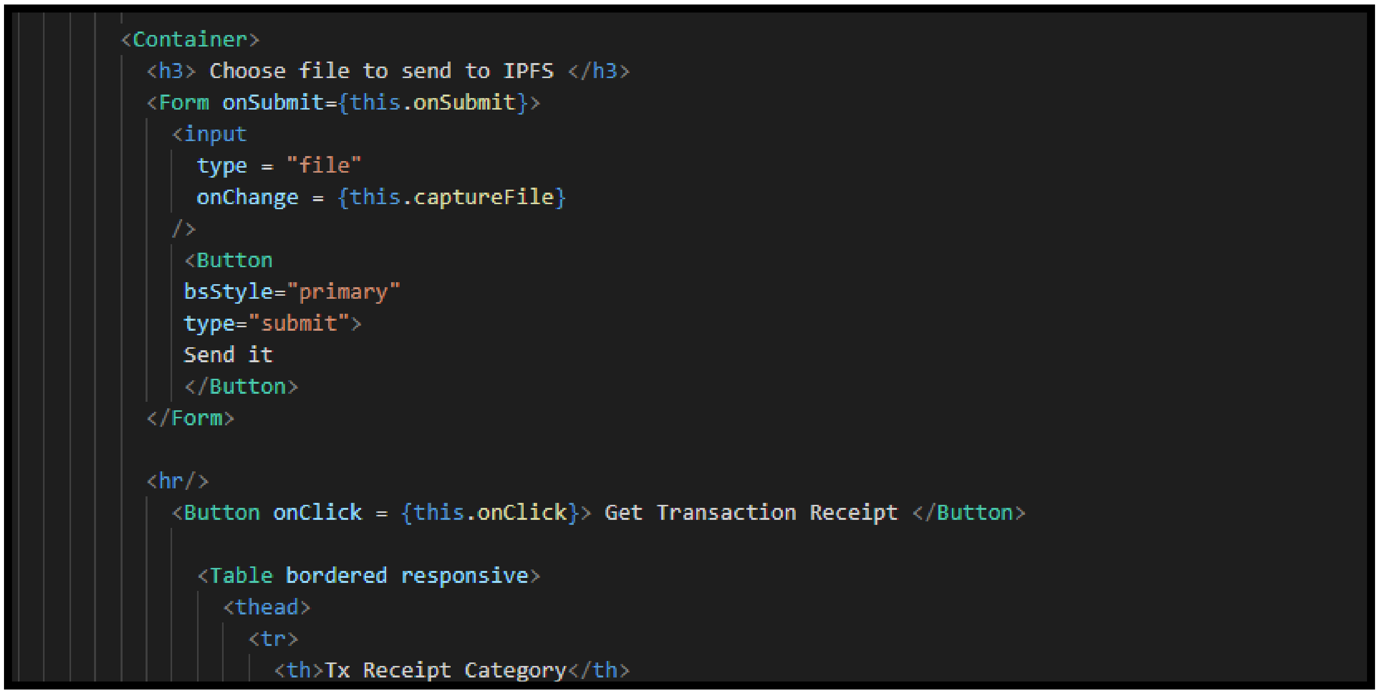
```
    onClick = async () => {

      try{
        this.setState({blockNumber:"waiting.."});
        this.setState({gasUsed:"waiting..."});

        // get Transaction Receipt in console on click
        // See: https://web3js.readthedocs.io/en/1.0/web3-eth.html#gettransactionreceipt
        await web3.eth.getTransactionReceipt(this.state.transactionHash, (err, txReceipt)=>{
          console.log(err,txReceipt);
          this.setState({txReceipt});
        }); //await for getTransactionReceipt

         this.setState({blockNumber: this.state.txReceipt.blockNumber});
         this.setState({gasUsed: this.state.txReceipt.gasUsed});
      } //try
    catch(error){
        console.log(error);
      } //catch
  } //onClick

    onSubmit = async (event) => {
```



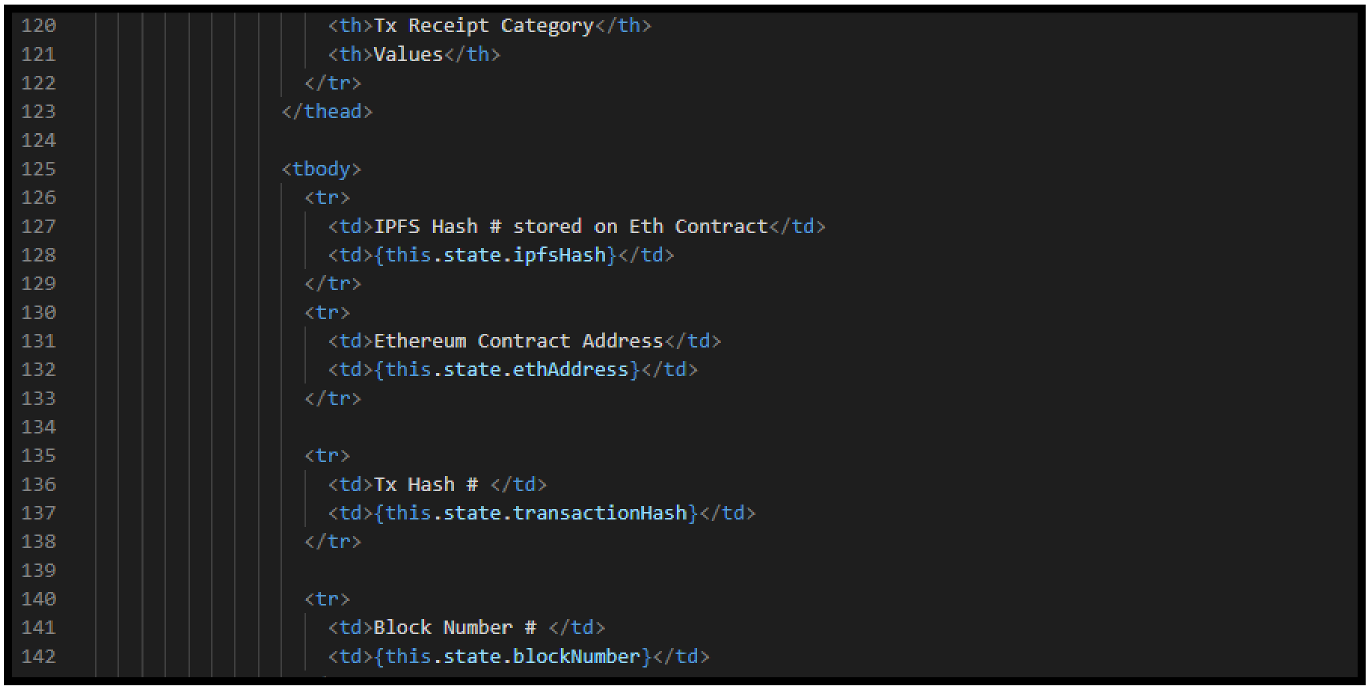
```
    onSubmit = async (event) => {
      event.preventDefault();

      //bring in user's metamask account address
      const accounts = await web3.eth.getAccounts();

      console.log('Sending from Metamask account: ' + accounts[0]);

      //obtain contract address from storehash.js
      const ethAddress=storehash.options.address;
      this.setState({ethAddress});

      //save document to IPFS,return its hash#, and set hash# to state
      //https://github.com/ipfs/interface-ipfs-core/blob/master/SPEC/FILES.md#add
      await ipfs.add(this.state.buffer, (err, ipfsHash) => {
        console.log(err,ipfsHash);
        //setState by setting ipfsHash to ipfsHash[0].hash
        this.setState({ ipfsHash:ipfsHash[0].hash });

        // call Ethereum contract method "sendHash" and .send IPFS hash to etheruem contract
        //return the transaction hash from the ethereum contract
        //see, this https://web3js.readthedocs.io/en/1.0/web3-eth-contract.html#methods-mymethod-send
```



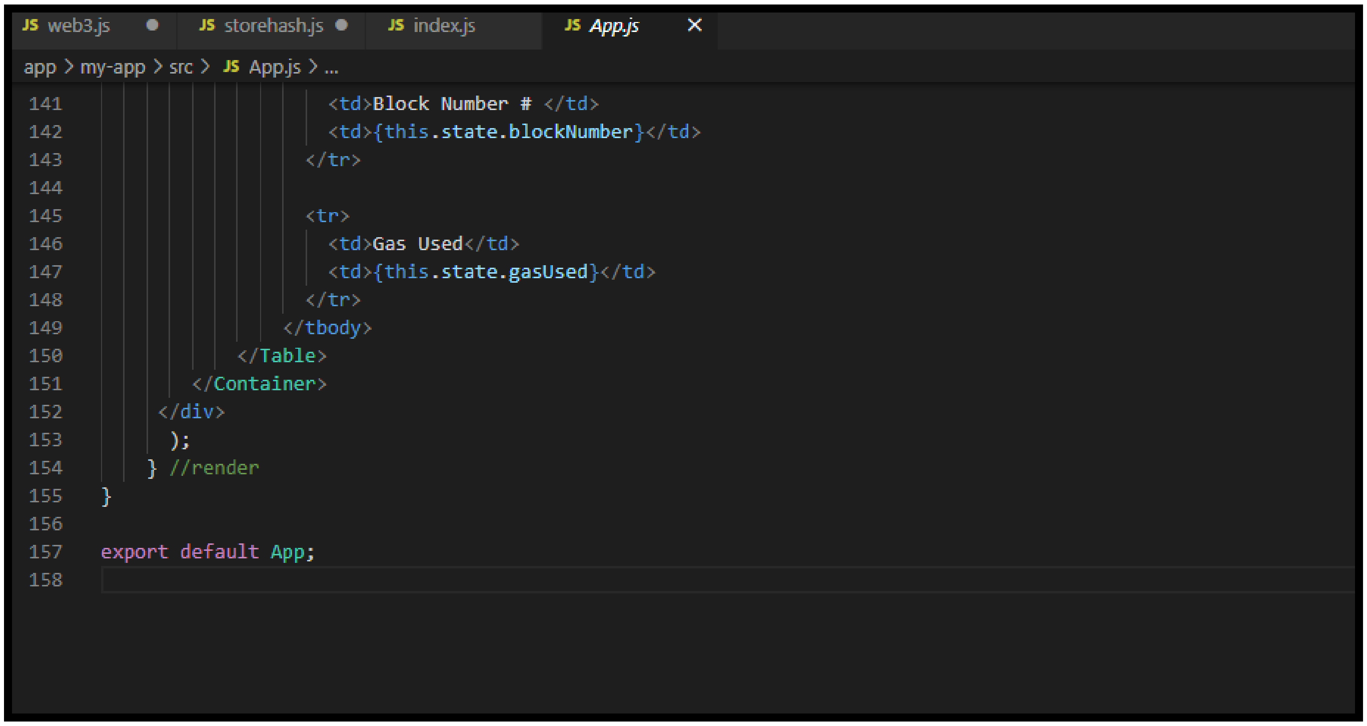
```
        this.setState({ ipfsHash:ipfsHash[0].hash });

        // call Ethereum contract method "sendHash" and .send IPFS hash to etheruem contract
        //return the transaction hash from the ethereum contract
        //see, this https://web3js.readthedocs.io/en/1.0/web3-eth-contract.html#methods-mymethod-send

        storehash.methods.sendHash(this.state.ipfsHash).send({
          from: accounts[0]
        }, (error, transactionHash) => {
          console.log(transactionHash);
          this.setState({transactionHash});
        }); //storehash
      }) //await ipfs.add
    }; //onSubmit

    render() {

      return (
        <div className="App">
          <header className="App-header">
            <h1> DECENTRALIZED EXPERIENCE/DEGREE VERIFICATION SYSTEM</h1>
          </header>
```



```
        <Container>
          <h3> Choose file to send to IPFS </h3>
          <Form onSubmit={this.onSubmit}>
            <input
              type = "file"
              onChange = {this.captureFile}
            />
             <Button
             bsStyle="primary"
             type="submit">
             Send it
             </Button>
          </Form>

          <hr/>
            <Button onClick = {this.onClick}> Get Transaction Receipt </Button>

              <Table bordered responsive>
                <thead>
                  <tr>
                    <th>Tx Receipt Category</th>
```



```
                    <th>Tx Receipt Category</th>
                    <th>Values</th>
                  </tr>
                </thead>

                <tbody>
                  <tr>
                    <td>IPFS Hash # stored on Eth Contract</td>
                    <td>{this.state.ipfsHash}</td>
                  </tr>
                  <tr>
                    <td>Ethereum Contract Address</td>
                    <td>{this.state.ethAddress}</td>
                  </tr>

                  <tr>
                    <td>Tx Hash # </td>
                    <td>{this.state.transactionHash}</td>
                  </tr>

                  <tr>
                    <td>Block Number # </td>
                    <td>{this.state.blockNumber}</td>
```



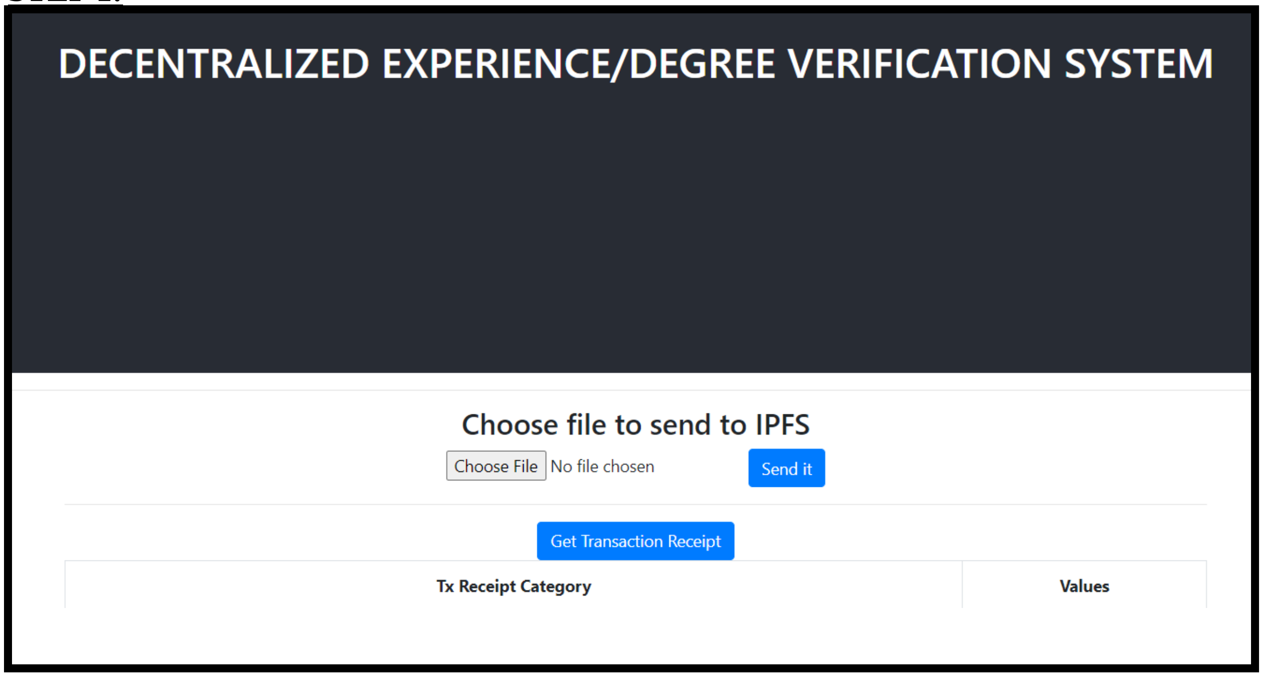
```
                    <td>Block Number # </td>
                    <td>{this.state.blockNumber}</td>
                  </tr>

                  <tr>
                    <td>Gas Used</td>
                    <td>{this.state.gasUsed}</td>
                  </tr>
                </tbody>
            </Table>
        </Container>
      </div>
        );
    } //render
}

export default App;

```

When files are being created and edited, then will be compiled simultaneously and if there is any issue that will be visible to us in terminal.

When files compiled without any error, you can access your application on the localhost:3000.

INTERACTING WITH APPLICATION:

STEP 1:



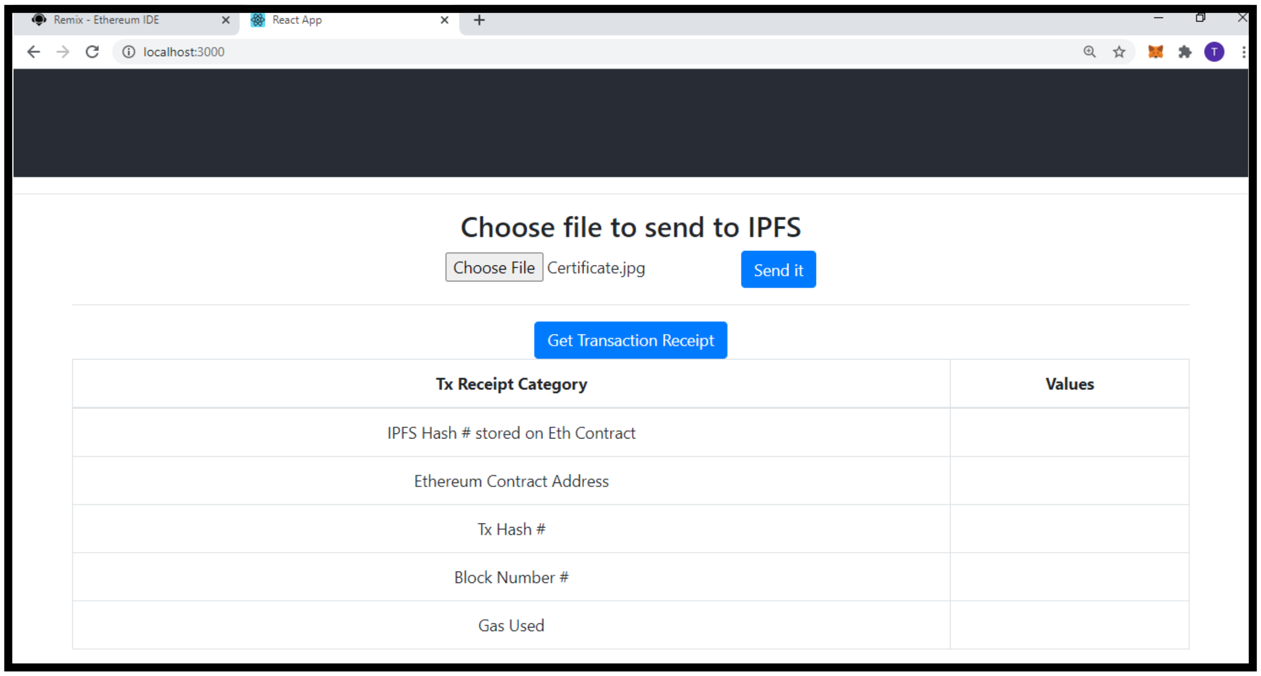
DECENTRALIZED EXPERIENCE/DEGREE VERIFICATION SYSTEM

Choose file to send to IPFS

Tx Receipt Category	Values

STEP 2:

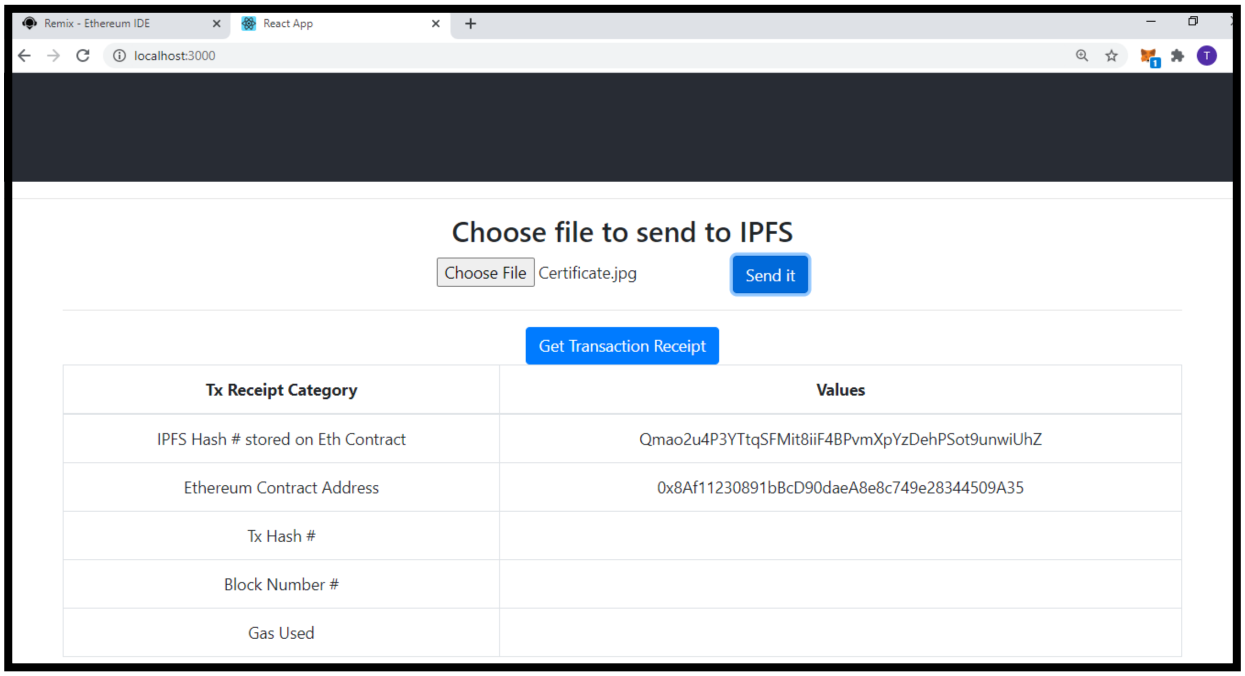
Now choose the certificate you want to upload and the send the file by using "Send it" button. In the window below a sample certificat has been uploaded but any type of file can be uploaded.



Choose file to send to IPFS

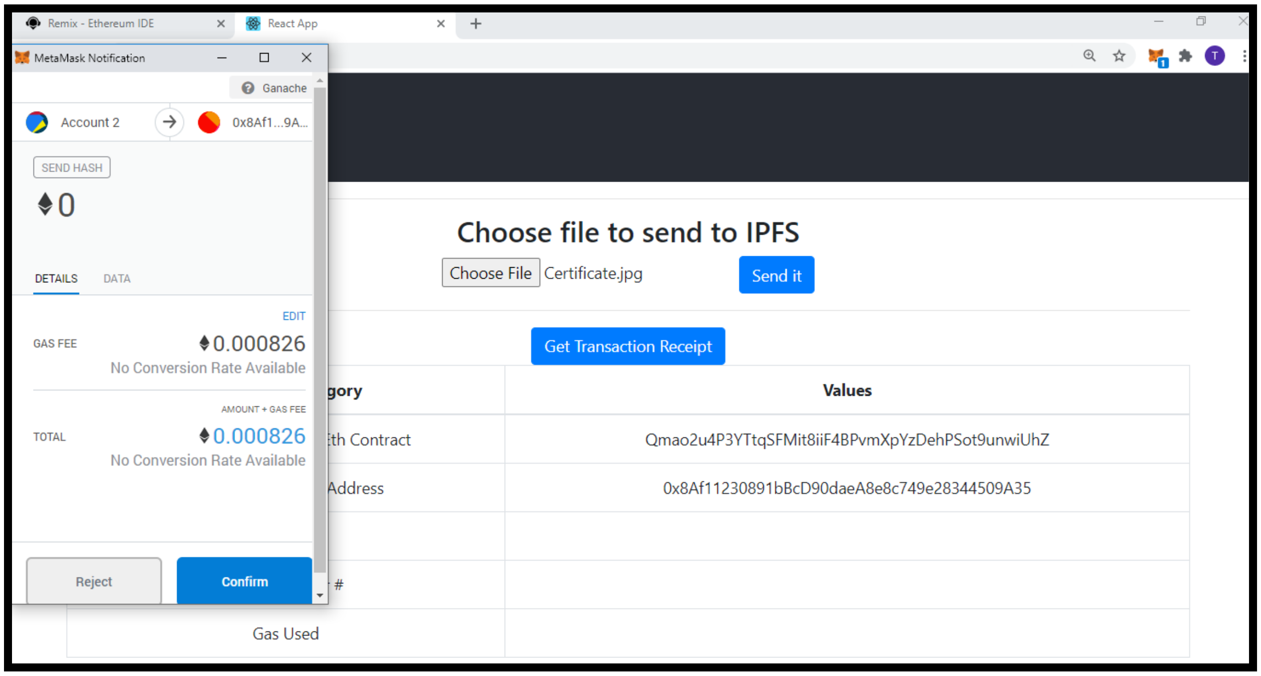
Tx Receipt Category	Values
IPFS Hash # stored on Eth Contract	
Ethereum Contract Address	
Tx Hash #	
Block Number #	
Gas Used	

STEP 3:
The smart contract address will be shown and an IPFS hash is generated which is displayed as shown below:



Tx Receipt Category	Values
IPFS Hash # stored on Eth Contract	Qmao2u4P3YTtqSFMit8iiF4BPvmXpYzDehPSot9unwiUhZ
Ethereum Contract Address	0x8Af11230891bBcD90daeA8e8c749e28344509A35
Tx Hash #	
Block Number #	
Gas Used	

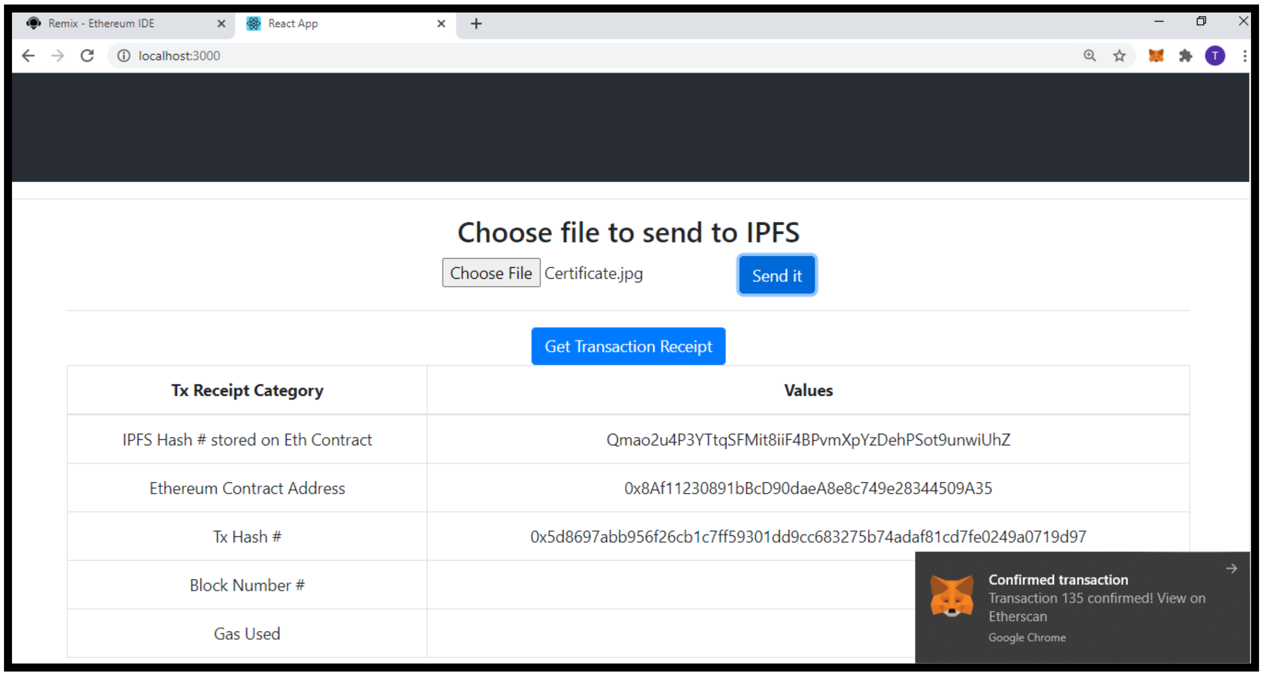
STEP 4:
A metamask notification will be displayed. Select "confirm" to execute the transaction.



...gory	Values
...th Contract	Qmao2u4P3YTtqSFMit8iiF4BPvmXpYzDehPSot9unwiUhZ
...Address	0x8Af11230891bBcD90daeA8e8c749e28344509A35
...#	
Gas Used	

STEP 5:

On successful execution of transaction, a transaction hash will be displayed.

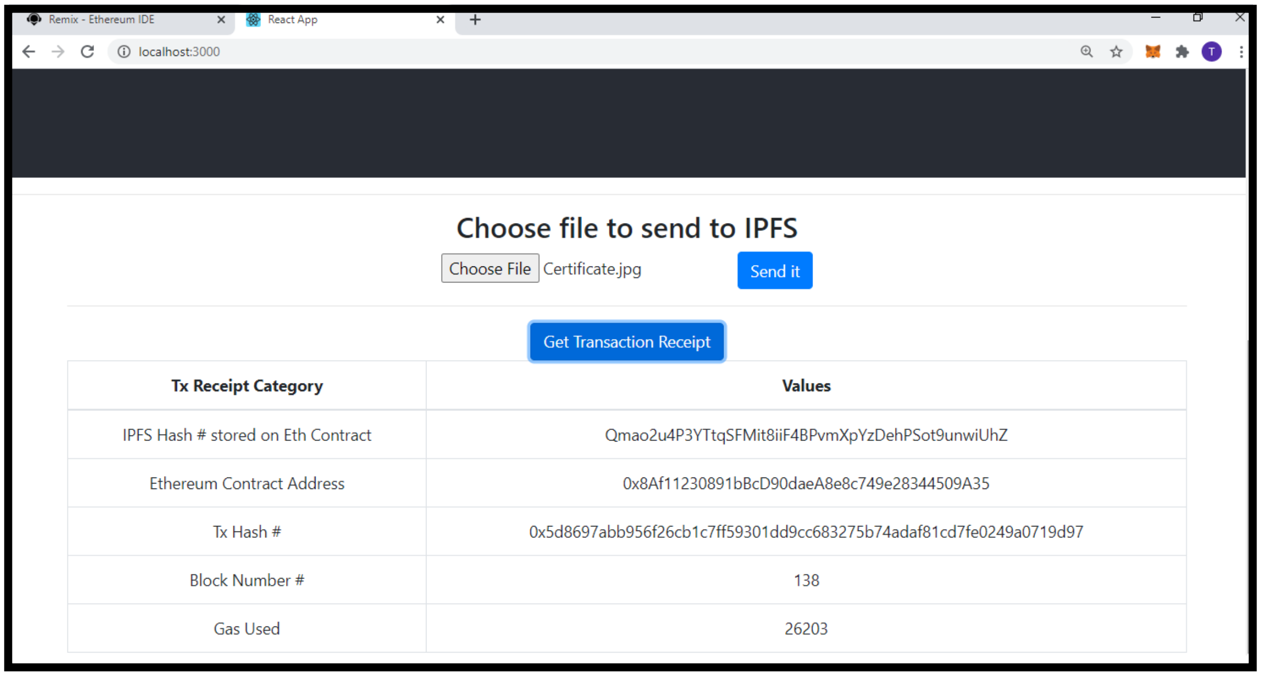


Tx Receipt Category	Values
IPFS Hash # stored on Eth Contract	Qmao2u4P3YTtqSFMit8iiF4BPvmXpYzDehPSot9unwiUhZ
Ethereum Contract Address	0x8Af11230891bBcD90daeA8e8c749e28344509A35
Tx Hash #	0x5d8697abb956f26cb1c7ff59301dd9cc683275b74adaf81cd7fe0249a0719d97
Block Number #	
Gas Used	

Confirmed transaction
Transaction 135 confirmed! View on Etherscan
Google Chrome

STEP 6:

Select “ Get Transaction Receipt” in order to get the details of the transaction.

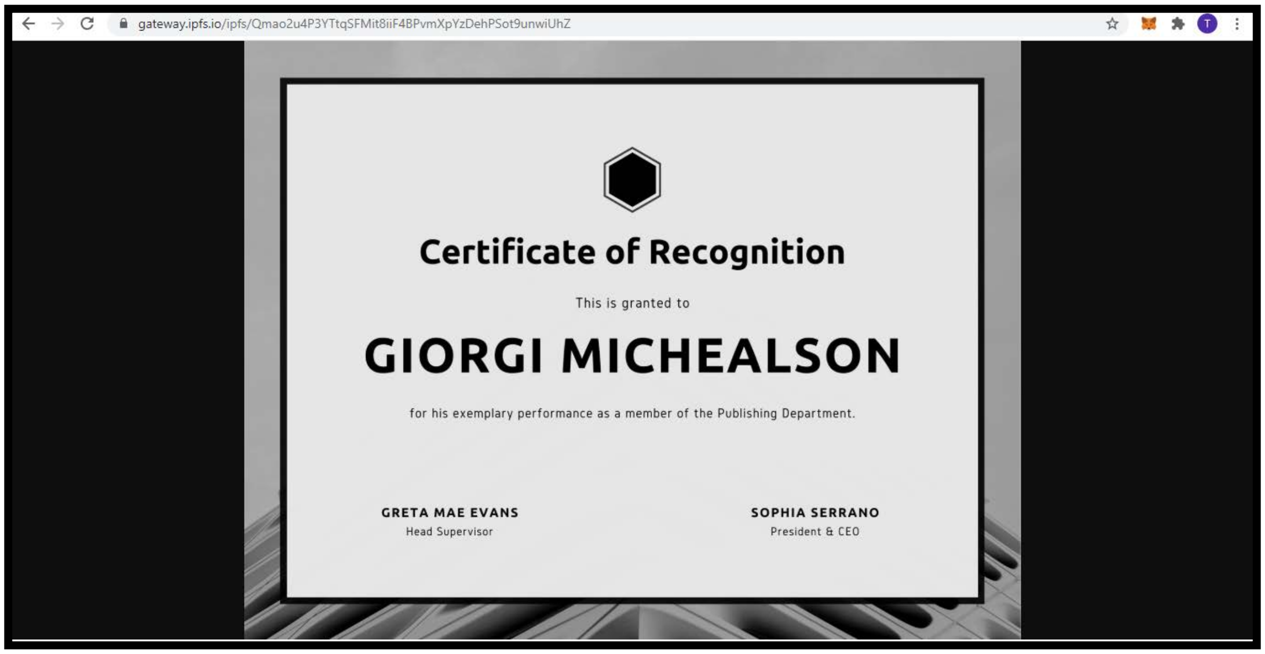


Choose file to send to IPFS

Tx Receipt Category	Values
IPFS Hash # stored on Eth Contract	Qmao2u4P3YTtqSFMit8iiF4BPvmXpYzDehPSot9unwiUhZ
Ethereum Contract Address	0x8Af11230891bBcD90daeA8e8c749e28344509A35
Tx Hash #	0x5d8697abb956f26cb1c7ff59301dd9cc683275b74adaf81cd7fe0249a0719d97
Block Number #	138
Gas Used	26203

VIEWING YOUR UPLOADED FILES:

The uploaded files can be accessed at https://gateway.ipfs.io/ipfs/ + your IPFS hash#.
https://gateway.ipfs.io/ipfs/Qmao2u4P3YTtqSFMit8iiF4BPvmXpYzDehPSot9unwiUhZ



Certificate of Recognition

This is granted to

GIORGI MICHEALSON

for his exemplary performance as a member of the Publishing Department.

GRETA MAE EVANS
Head Supervisor

SOPHIA SERRANO
President & CEO

This page is left intentionally blank.