GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

CERTIFICATE VERIFICATION SYSTEM USING SMART CONTRACT ON BLOCKCHAIN

A Report for the Evaluation 3 of Project 2

Submitted by

KUMAR MAYANK

(1613108001 / 16SCSE108001)

in partial fulfillment for the award of the degree

of

Bachelor of Technology

IN

Computer Science and Engineering With Specialization of Cloud Computing and Virtualization

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Dr. SPS Chauhan

(Assoc. Prof)

APRIL / MAY- 2020

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1.	Abstract	1
2.	Introduction	2
3.	Existing System	5
4.	Proposed system	6
5.	Implementation or architecture diagrams	8
6.	Output / Result / Screenshot	10
7.	Conclusion/Future Enhancement	15
8.	References	16

1. Abstract

This Report cover analysis of a Certificate verification system that allows an organization to store important certificate in a de-centralized network storage which makes certificates secure and unmodified.

DAPPs or Decentralized Application does not have a single centralized authority which governs its working, instead there are a number of decentralized authorities on which a DAPP works. Also, using blockchain comes with proof of work and miners. Moreover the data on the blockchain is immutable and distributed so it's certainly impossible to hack the application.
In this Certificate verification System we use Ganache which provides us our personal local blockchain network which we can use to develop our blockchain application. It also gives temporary test accounts with ethereum which we can use to run the app.One of the best way to implement a decentralized application is using Ethereum Blockchain network as Ethereum Blockchain have methods to deploy Smart Contracts.

A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties.
There are a lot of fake certificate rolls out every year and it's very hard to verify it .This system provides a more secure way to handle the certificate in a de-Centralized Storage and to verify the Documents either they are genuine or not.

2. Introduction

Overall Description

Blockchain technology that shines like a star after the entrance and widespread acceptance of Bitcoin , the very first cryptocurrency in peoples' everyday life, has become a trending topic in today's software world. At the beginning, Blockchain was only used for monetary transactions and trade, but studies have started to suggest that it can be used in many more areas over time, because there is a high degree of transparency in this system. For example, in Bitcoin, since the wallets are in a distributed structure, the total amount of coins and instant transaction volume in the world can be followed momentarily and clearly. There is no need for a central authority to approve or complete the operations on this P2P-based system. Because of that, not only the money transfers but also all kinds of structural information can be kept in this distributed chain, and with the help of some crypto logical methods, the system can be maintained securely. Like people's assets, marriage certificates, bank account books, medical information, etc., a lot of information can be recorded with this system with relevant modifications. Ethereum coin (Ether), another cryptocurrency with multipurpose development environments, which emerged a few years after Bitcoin, distinguishes the blockchain in a real sense, revealing that this technology can produce software that can hold information that is structured as described above. The software programs enforced by smart contracts are written into the blockchain and are immutable. They cannot be (illegally) removed nor manipulated once written. Hence, they can work properly, autonomously and transparently forever, without any external stimuli.

Blockchain is a distributed, immutable, incontrovertible, public ledger. This new technology has three main features:

Immutability: Any proposed "new block" to the ledger must reference the previous version of the ledger. This creates an immutable chain, which is where the blockchain gets its name from, and prevents tampering with the integrity of the previous entries.

(i) Verifiability: The ledger is decentralized, replicated and distributed over multiple locations. This ensures high availability (by eliminating a single point of failure) and provides third-party verifiability as all nodes maintain the consensus version of the ledger.

(ii) Distributed Consensus: A distributed consensus protocol to determine who can append the next new transaction to the ledger. A majority of the network nodes must reach a consensus before any new proposed block of entries becomes a permanent part of the ledger. These features are in part achieved through advanced cryptography, providing a security level greater than any previously known record-keeping system.

An idea of Open source application for managing certificate

- The idea of project is to build a open source platform to manage certificate in more easy and secure way.
- The platform will provide a unique area to store and verify for fake certificates.
- Free and open source will let organization to choose one single platform for storing certificate.
- A de-centralized app running on blockchain will provide a more secure way to store the certificates.

Early Attempts:

- Previously, the certificates are managed by individual organization with use of physical papers that cost more and too difficult to handle.
- The large number of certificate also makes its tuff to store and distribute to the candidate.
- The paper sheet methods are not so safe as any physical damage let full spoil of certificate.

The project try to make a unique platform where we can verify a certificate is valid or a fake one. This idea needs good enrollment of firms and organization to issue certificate through a unique open source platform. Nothing is permanently Secured it requires a team to build time-to-time Security patches.

Implementation of project By Smart Contract :

- ‣ A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties.

- ‣ One of the best things about the blockchain is that, because it is a decentralized system that exists between all permitted parties, there's no need to pay intermediaries (Middlemen) and it saves you time and conflict. Blockchains have their problems, but they are rated, undeniably, faster, cheaper, and more secure than traditional systems, which is why banks and governments are turning to them.

3. Existing System

Early Attempts for this was :

- Previously, the certificates are managed by individual organization with use of physical papers that cost more and too difficult to handle.

- The large number of certificate also makes its tuff to store and distribute to the candidate.

- The paper sheet methods are not so safe as any physical damage let full spoil of certificate.

The project try to make a unique platform where we can verify a certificate is valid or a fake one. This idea needs good enrollment of firms and organization to issue certificate through a unique open source platform. Nothing is permanently Secured it requires a team to build time-to-time Security patches.

4. Proposed Model

A Decentralized Voting Application Using Ethereum Blockchain

The tech world is one of the most dynamic segments in the whole universe. One moment, the world is behind a technology and the next moment, suddenly, the technology becomes obsolete. Similar is the case with the app world, too. Numerous tech stacks, frameworks, and languages are available to develop an app but still, developers are not confident about a single framework that can offer the best results.

As the world is adjusting to conventional apps, the whole ecosystem is also evolving. dApps or decentralized applications are a novel breed of applications that are not controlled or owned by a single authority, cannot be shut-off or cannot have a downtime.

dApps: The Ultimate Open Source Revolution

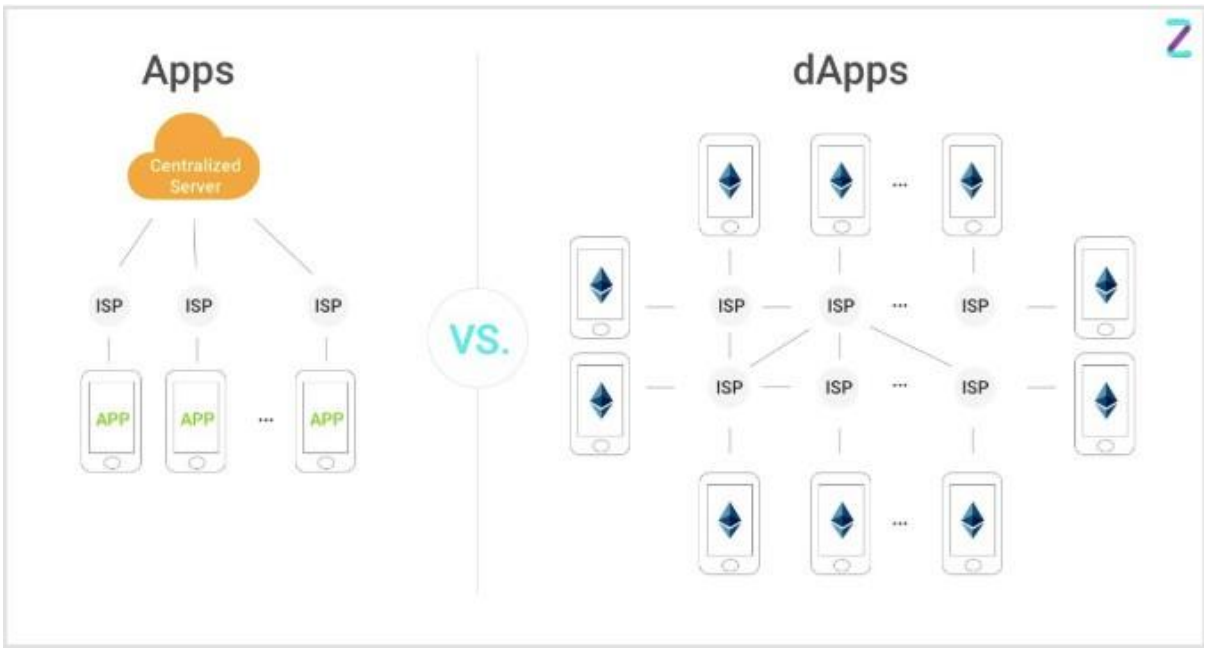
The dApp concept is still in its nascent stage. Explaining the same in a single line is tough because no specific definition seems to fit all the attributes that make an application a decentralized app. As dApps, an application is required to exhibit the following four characteristics:

- ***Open Source:*** The first and foremost attribute is that such apps should make their core source code available to everyone. As the core characteristic of dApps is autonomy and unanimous consensus, essentially the changes must be decided by all or the majority of the users. Also, the code should be available to everyone for checking out.
- ***Decentralized Nature:*** As the name suggests, decentralized applications stores everything on a decentralized blockchain or any cryptographic technology to save the app from perils of centralized authority and emphasize on autonomous nature.
- ***Incentivization:*** As the app is based on the decentralized blockchain, the validators of the records on the network must be rewarded/incentivized with cryptographic tokens or any form of digital asset that has value.
- ***Algorithm:*** Decentralized app needs to have a consensus mechanism that portrays proof of value in the cryptographic system. Essentially, this endows value to the cryptographic token and creates a consensus protocol that users agree upon to generate valuable crypto tokens.

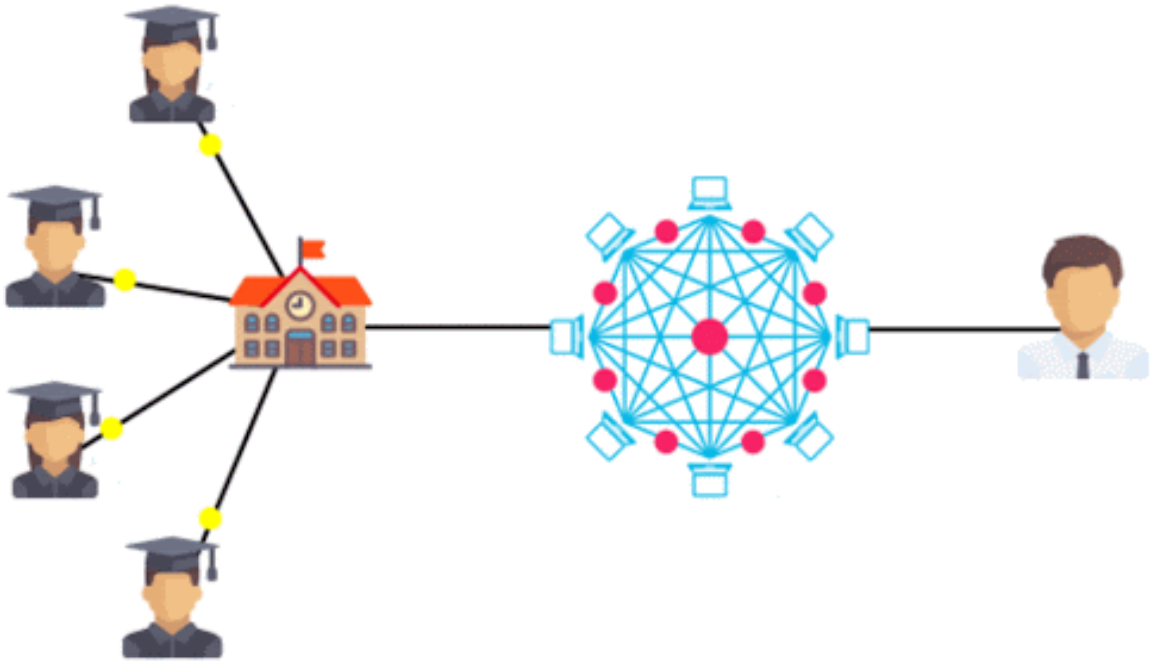
Now that we know the characteristics, we can try to fit it into a definition that will help us identify real-life examples. Essentially, dApp is an application that runs on decentralized P2P network governed by all the members and not a single central authority.

How dApps fit in the real world?

Utilizing the definition derived above, we found out that the first known dApp in the world was the Bitcoin. Popular as an apex cryptocurrency, bitcoin solves the centralization issue and gives users the power to perform transactions without any middleman or central authority via a self-sustaining public ledger. Coming to the use case of decentralized applications, we can classify these apps based on the scenario they can be infused into.



dApp using Ethereum Blockchain



How Dapp is accessed by various users.

5. Implementation

The implementation of the dApp requires following dependencies to complete the working of the certificate verfiaction system on Ethereum Blockchain.

- Truffle Suite
- Ganache
- Solidity
- JavaScript
- Web3.js
- MetaMask
- Remix IDE

Ganache

Ganache is a personal blockchain for Ethereum development you can use to deploy contracts, develop your applications, and run tests.

SOLIDITY

› Solidity is a contract-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM).

› Soidity is statically typed, supports inheritance and complex user-defined types among other features.

Truffle Suite

‣ A world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier. With Truffle, you get:

‣ Built-in smart contract compilation, linking, deployment and binary management.

‣ Automated contract testing for rapid development.

‣ Scriptable, extensible deployment & migrations framework.

Remix IDE

Remix is a Solidity IDE that's used to write, compile and debug Solidity code. Solidity is a high-level, contract-oriented programming language for writing smart contracts. It was influenced by popular languages such as C++, Python and JavaScript.

Implementation Steps :

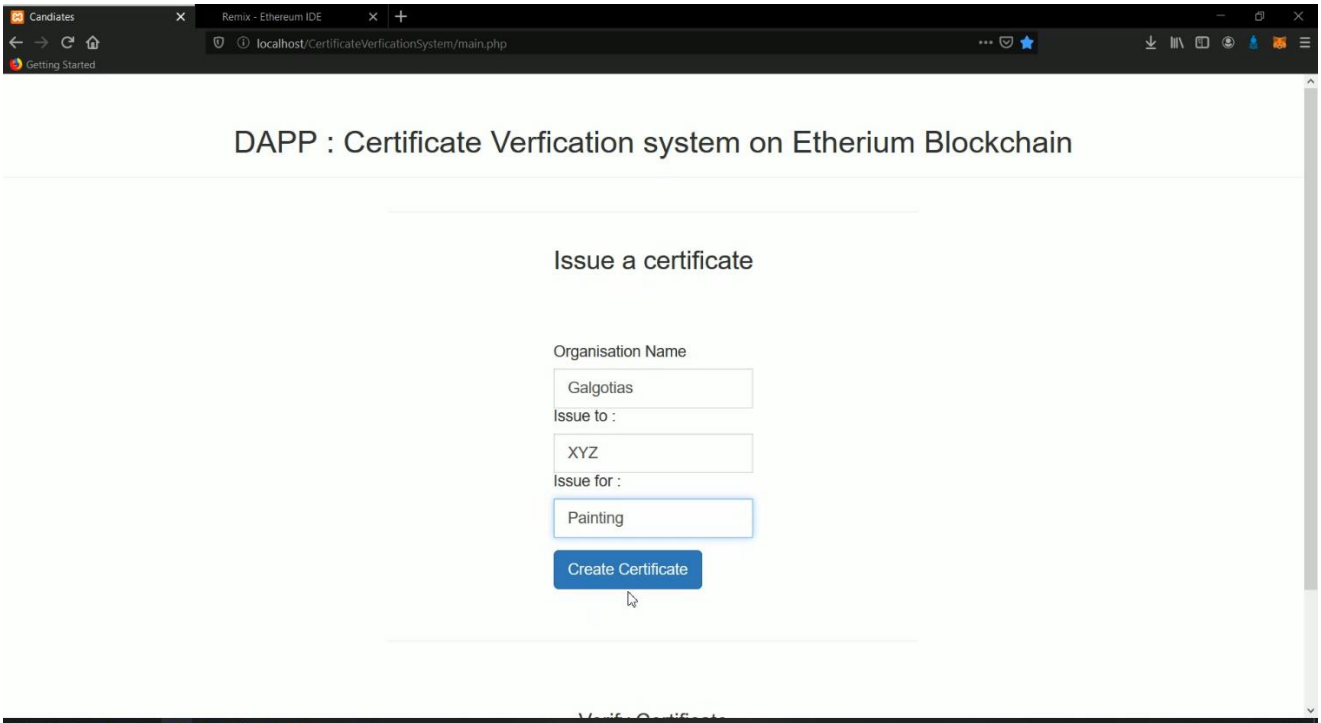
Setting up environment:

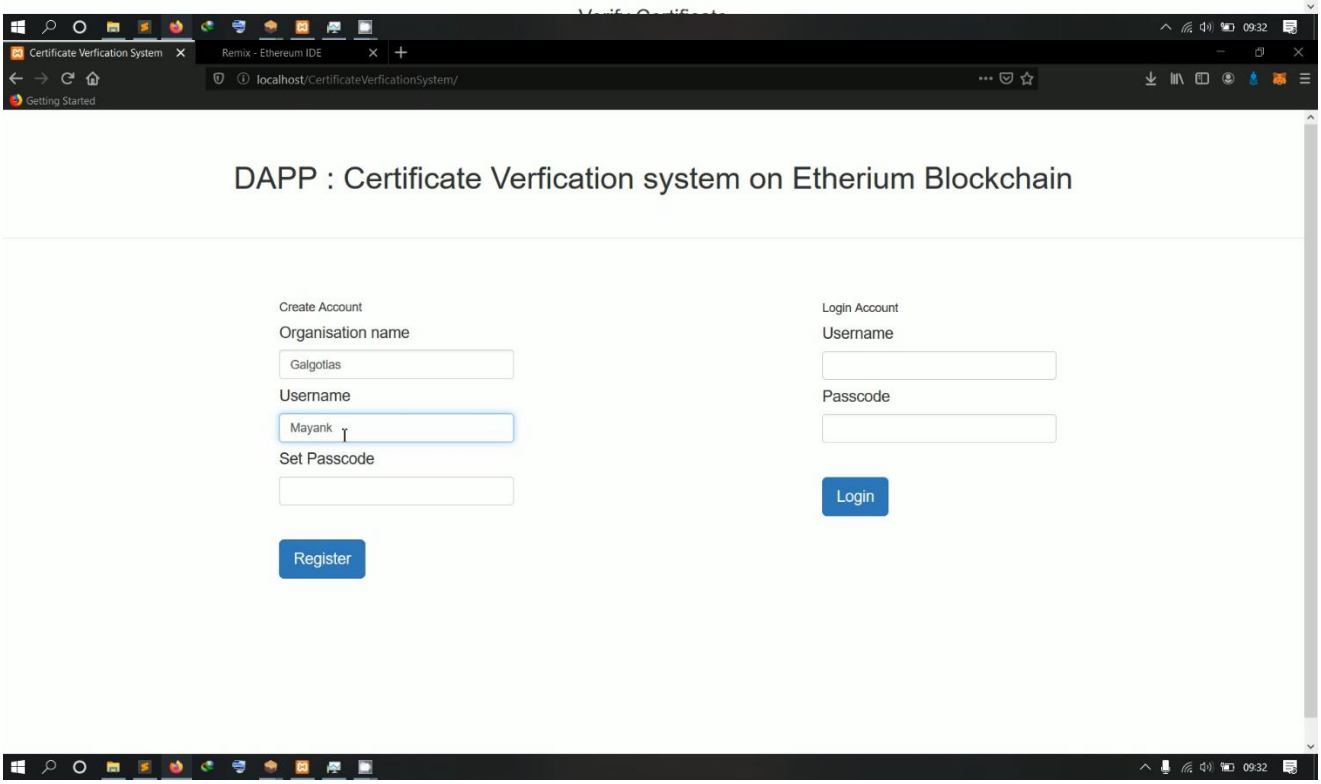
1. Run Ganche to create a local server of blockchain on pc.
2. Goto IDE and compile and deploy the contract
3. Copy and paste the address codes in project code.
4. Run a php server to host dapp webpage.

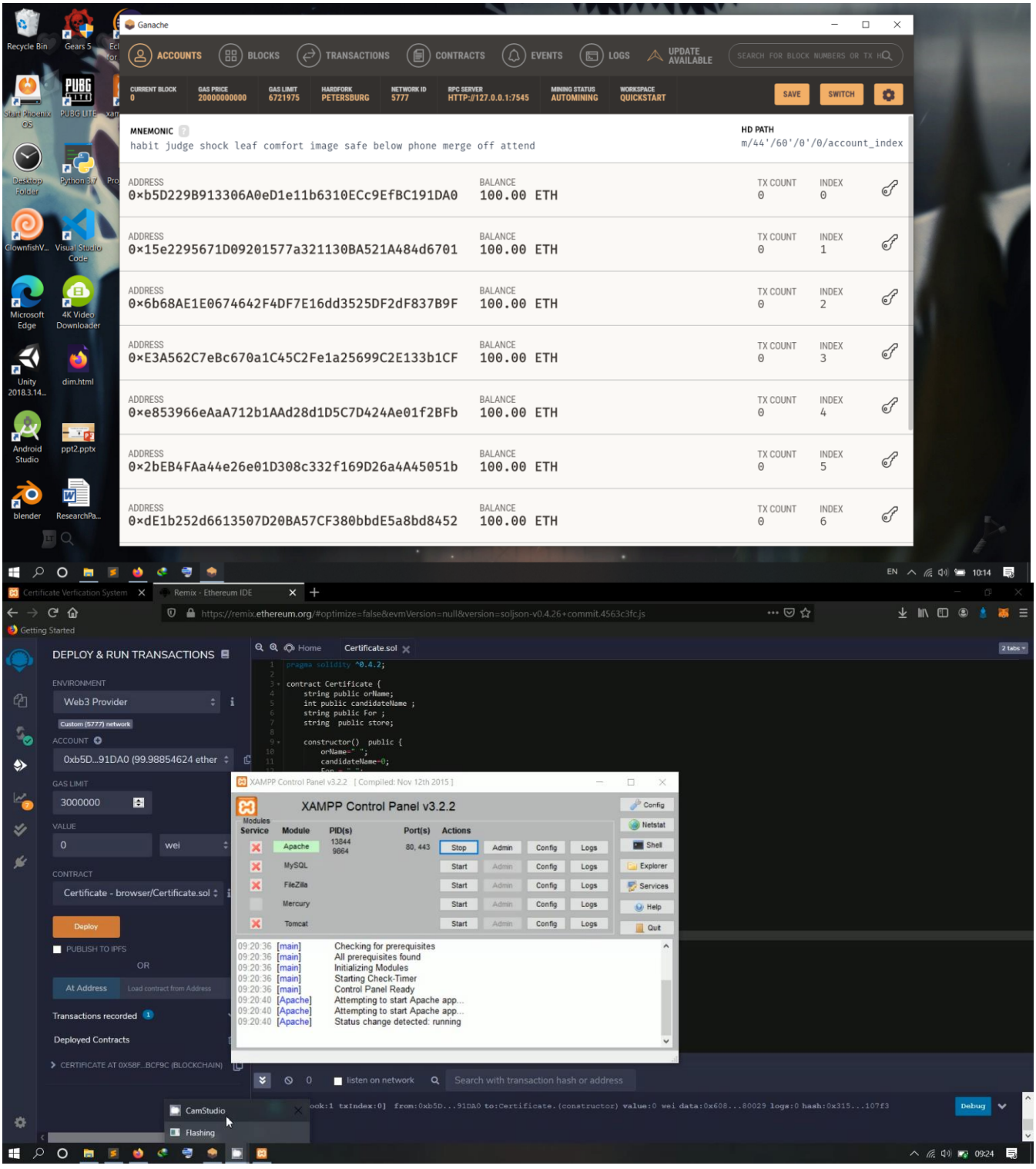
Running the application:

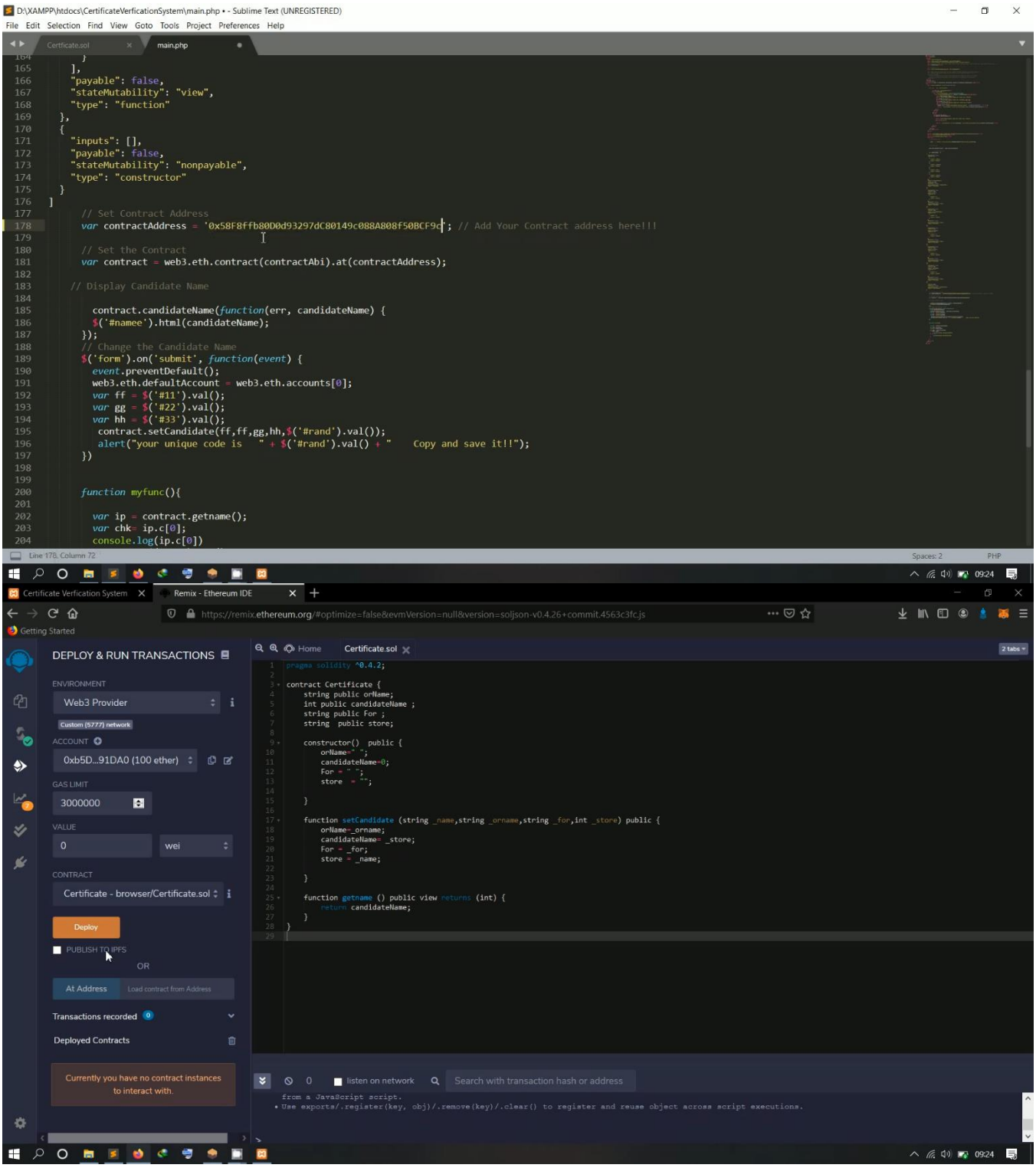
1. Run the Dapp webpage by hosted local server.
2. Create an account for your organization.
3. Issue a certificate by first form
4. Verify a certificate by verification form.

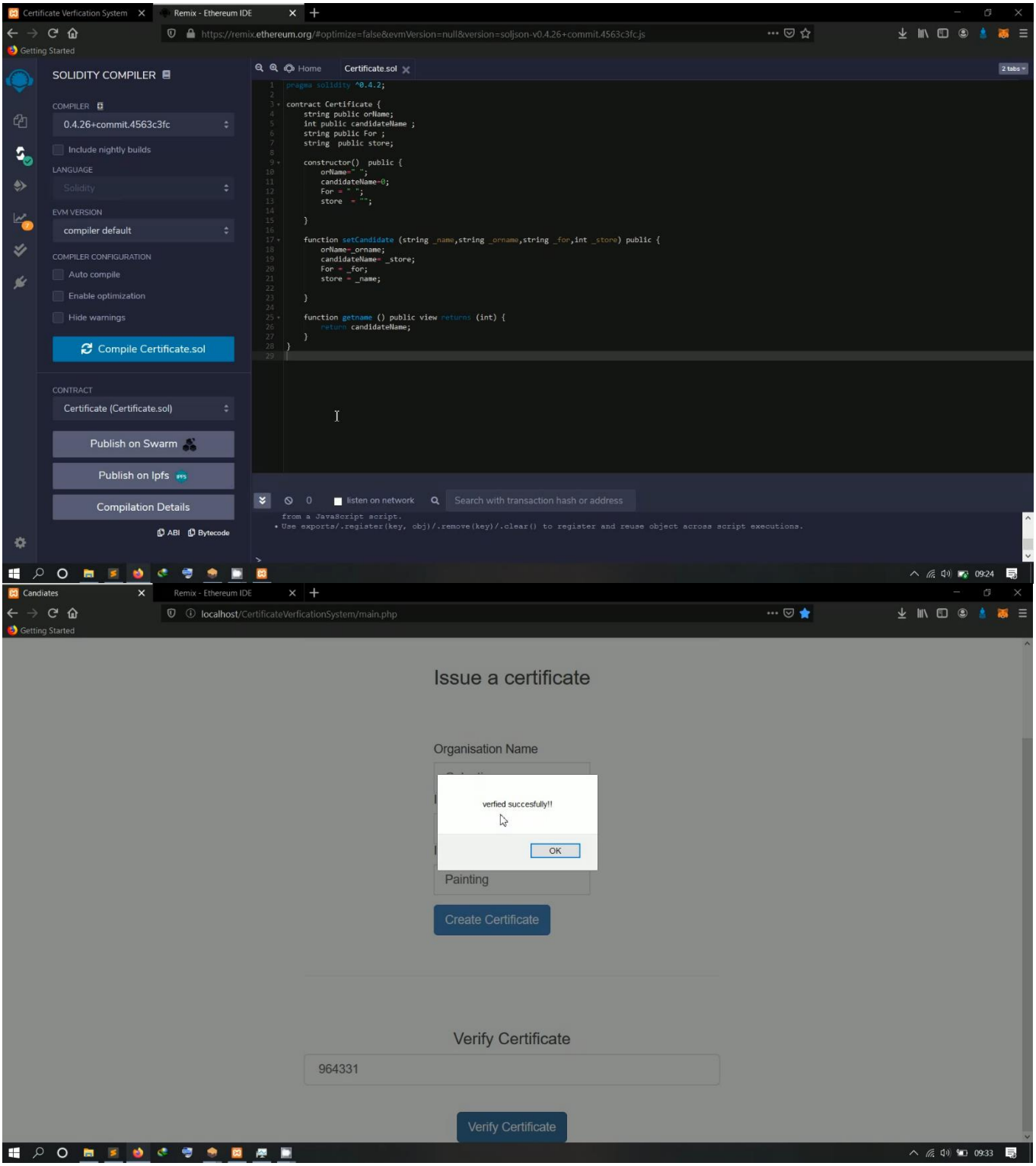
6. Screenshots and Result

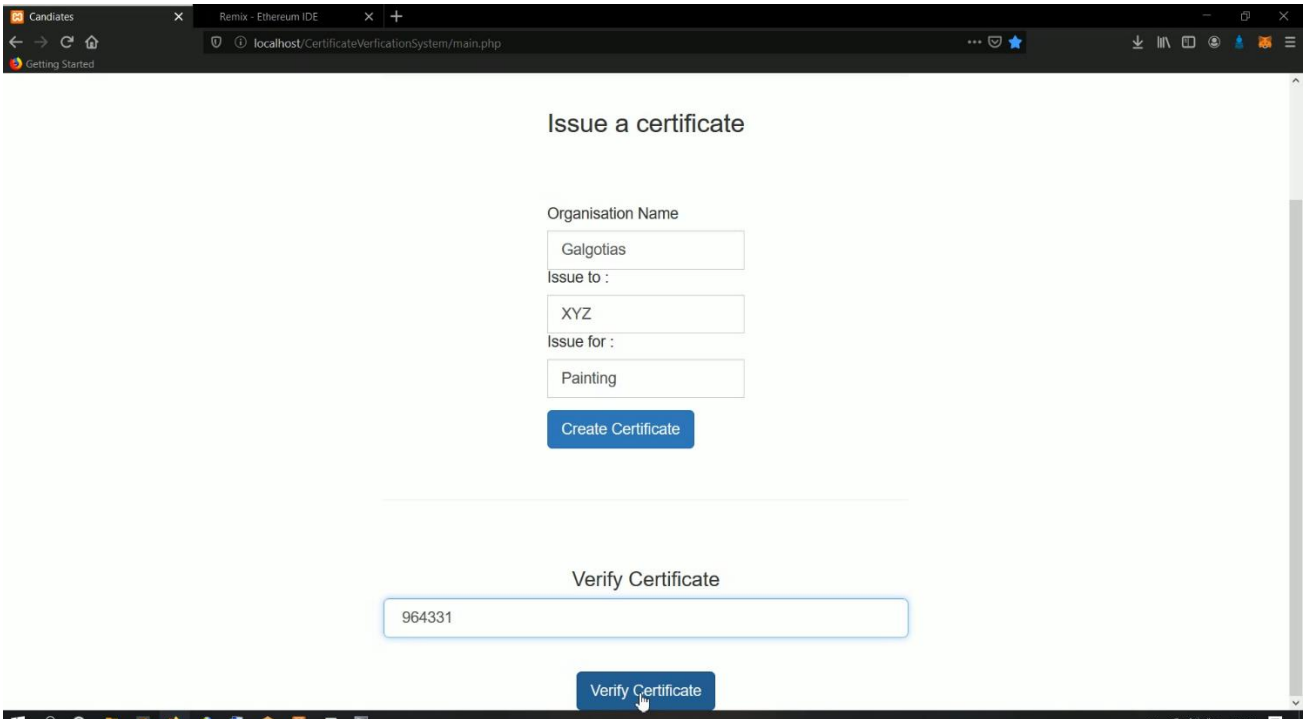


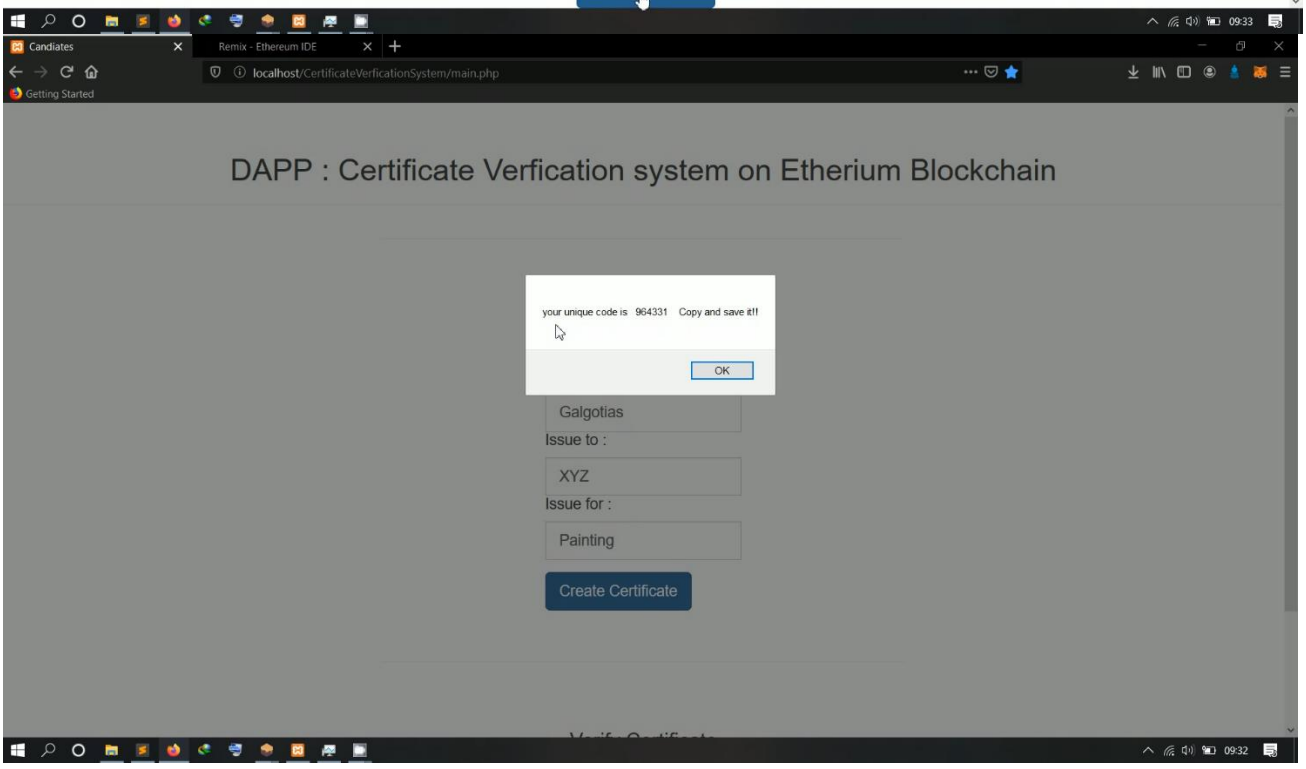



Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
UPDATE AVAILABLE
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK PETERSBURG
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART
SAVE
SWITCH
MNEMONIC
habit judge shock leaf comfort image safe below phone merge off attend
HD PATH
m/44'/60'/0'/0/account_index
ADDRESS 0x5D229B913306A0eD1e11b6310ECc9EfBC191DA0 BALANCE 100.00 ETH TX COUNT 0 INDEX 0
ADDRESS 0x15e2295671D09201577a321130BA521A484d6701 BALANCE 100.00 ETH TX COUNT 0 INDEX 1
ADDRESS 0x6b68AE1E0674642F4DF7E16dd3525DF2dF837B9F BALANCE 100.00 ETH TX COUNT 0 INDEX 2
ADDRESS 0xE3A562C7eBc670a1C45C2Fe1a25699C2E133b1CF BALANCE 100.00 ETH TX COUNT 0 INDEX 3
ADDRESS 0xe853966eAaA712b1AAd28d1D5C7D424Ae01f2BFb BALANCE 100.00 ETH TX COUNT 0 INDEX 4
ADDRESS 0x2bEB4FAa44e26e01D308c332f169D26a4A45051b BALANCE 100.00 ETH TX COUNT 0 INDEX 5
ADDRESS 0xdE1b252d6613507D20BA57CF380bbdE5a8bd8452 BALANCE 100.00 ETH TX COUNT 0 INDEX 6
Recycle Bin
Gears 5
Start Phoenix OS
PUBG LITE
Desktop Folder
Python 3.7
ClownfishV...
Visual Studio Code
Microsoft Edge
4K Video Downloader
Unity 2018.3.14...
dim.html
Android Studio
ppt2.pptx
blender
ResearchPa...
Certificate Verfication System
Remix - Ethereum IDE
https://remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js
Getting Started
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Web3 Provider
Custom (5777) network
ACCOUNT
0xb5D...91DA0 (99.98854624 ether
GAS LIMIT
3000000
VALUE
0
wei
CONTRACT
Certificate - browser/Certificate.sol
Deploy
PUBLISH TO IPFS
OR
At Address
Load contract from Address
Transactions recorded 1
Deployed Contracts
CERTIFICATE AT 0X58F...BCF9C (BLOCKCHAIN)
Home
Certificate.sol
pragma solidity ^0.4.2;
contract Certificate {
string public orName;
int public candidateName ;
string public For ;
string public store;
constructor() public {
orName=" ";
candidateName=0;
XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Modules
Service Module PID(s) Port(s) Actions
Apache 13844 9864 80, 443 Stop Admin Config Logs
MySQL Start Admin Config Logs
FileZilla Start Admin Config Logs
Mercury Start Admin Config Logs
Tomcat Start Admin Config Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit
09:20:36 [main] Checking for prerequisites
09:20:36 [main] All prerequisites found
09:20:36 [main] Initializing Modules
09:20:36 [main] Starting Check-Timer
09:20:36 [main] Control Panel Ready
09:20:40 [Apache] Attempting to start Apache app...
09:20:40 [Apache] Attempting to start Apache app...
09:20:40 [Apache] Status change detected: running
listen on network
Search with transaction hash or address
CamStudio
Flashing
Debug


D:\XAMPP\htdocs\CertificateVerficationSystem\main.php • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Certficate.sol
main.php
"payable": false,
"stateMutability": "view",
"type": "function"
"inputs": [],
"payable": false,
"stateMutability": "nonpayable",
"type": "constructor"
// Set Contract Address
var contractAddress = '0x58F8ffb80D0d93297dC80149c088A808f50BCF9c'; // Add Your Contract address here!!!
// Set the Contract
var contract = web3.eth.contract(contractAbi).at(contractAddress);
// Display Candidate Name
contract.candidateName(function(err, candidateName) {
$('#namee').html(candidateName);
});
// Change the Candidate Name
$('form').on('submit', function(event) {
event.preventDefault();
web3.eth.defaultAccount = web3.eth.accounts[0];
var ff = $('#11').val();
var gg = $('#22').val();
var hh = $('#33').val();
contract.setCandidate(ff,ff,gg,hh,$('#rand').val());
alert("your unique code is " + $('#rand').val() + " Copy and save it!!");
})
function myfunc(){
var ip = contract.getname();
var chk= ip.c[0];
console.log(ip.c[0])
Line 178, Column 72
Spaces: 2
PHP
09:24
Certificate Verfication System
Remix - Ethereum IDE
https://remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js
Getting Started
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Web3 Provider
Custom (5777) network
ACCOUNT
0xb5D...91DA0 (100 ether)
GAS LIMIT
3000000
VALUE
0
wei
CONTRACT
Certificate - browser/Certificate.sol
Deploy
PUBLISH TO IPFS
OR
At Address
Load contract from Address
Transactions recorded
Deployed Contracts
Currently you have no contract instances to interact with.
Home
Certificate.sol
2 tabs
pragma solidity ^0.4.2;
contract Certificate {
string public orName;
int public candidateName ;
string public For ;
string public store;
constructor() public {
orName=" ";
candidateName=0;
For = " ";
store = "";
}
function setCandidate (string _name,string _orname,string _for,int _store) public {
orName=_orname;
candidateName= _store;
For = _for;
store = _name;
}
function getname () public view returns (int) {
return candidateName;
}
}
0
listen on network
Search with transaction hash or address
from a JavaScript script.
Use exports/.register(key, obj)/.remove(key)/.clear() to register and reuse object across script executions.
09:24


Certificate Verfication System
Remix - Ethereum IDE
https://remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.4.26+commit.4563c3fc.js
Getting Started
SOLIDITY COMPILER
COMPILER
0.4.26+commit.4563c3fc
Include nightly builds
LANGUAGE
Solidity
EVM VERSION
compiler default
COMPILER CONFIGURATION
Auto compile
Enable optimization
Hide warnings
Compile Certificate.sol
CONTRACT
Certificate (Certificate.sol)
Publish on Swarm
Publish on Ipfs
Compilation Details
ABI
Bytecode
Home
Certificate.sol
2 tabs
pragma solidity ^0.4.2;
contract Certificate {
string public orName;
int public candidateName ;
string public For ;
string public store;
constructor() public {
orName=" ";
candidateName=0;
For = " ";
store = "";
}
function setCandidate (string _name,string _orname,string _for,int _store) public {
orName=_orname;
candidateName= _store;
For = _for;
store = _name;
}
function getname () public view returns (int) {
return candidateName;
}
}
0
listen on network
Search with transaction hash or address
from a JavaScript script.
Use exports/.register(key, obj)/.remove(key)/.clear() to register and reuse object across script executions.
09:24
Candiates
Remix - Ethereum IDE
localhost/CertificateVerficationSystem/main.php
Getting Started
Issue a certificate
Organisation Name
verfied succesfully!!
OK
Painting
Create Certificate
Verify Certificate
964331
Verify Certificate
09:33


Candiates
Remix - Ethereum IDE
localhost/CertificateVerficationSystem/main.php
Getting Started
Issue a certificate
Organisation Name
Galgotias
Issue to :
XYZ
Issue for :
Painting
Create Certificate
Verify Certificate
964331
Verify Certificate


Candiates
Remix - Ethereum IDE
localhost/CertificateVerficationSystem/main.php
Getting Started
DAPP : Certificate Verfication system on Etherium Blockchain
your unique code is 964331 Copy and save it!!
OK
Galgotias
Issue to :
XYZ
Issue for :
Painting
Create Certificate

7. Conclusion and future enhancement

We were able to successfully implement the above discussed idea and have a working model of the dApp for certificate verification system. In the developed application a user can currently use his/her Ethereum wallet to save his certificate in de-centralized storage.

Future Scope of the Project :

- The Project can help the organization to issue and verify certificates in an more easy and secured way.
- This can reduce the paper work that is good for environment.
- It also reduce the man power and provide a very effective de-centralized application to handle certificate in an ease manner.

8. References

- **https://ethereum.org/**
- https://www.trufflesuite.com/
- https://www.trufflesuite.com/ganache
- https://solidity.readthedocs.io/en/v0.5.3