A REPORT

ON

STORING EDUCATIONAL CERTIFICATES USING BLOCKCHAIN

BY

TANYA PRASAD	**2018B5A71054H**
RAJVI SAMPAT	**2018B4A70820G**
ROHAN SINHA	**2018A7PS0417H**

AT

NATIONAL INFORMATICS CENTRE(NIC)

A Practice School-1 Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(JUNE,2020)

A REPORT

ON

STORING EDUCATIONAL CERTIFICATES USING BLOCKCHAIN

BY

TANYA PRASAD	**2018B5A71054H**	**PHYSICS + COMPUTER SCIENCE**
RAJVI SAMPAT	**2018B4A70830G**	**MATHEMATICS+COMPUTER SCIENCE**
ROHAN SINHA	**2018A7PS0417H**	**COMPUTER SCIENCE**

Prepared in partial fulfilment of the

Practice School-1 Course Nos.

BITS C221/BITS C231/BITS C241

AT

NATIONAL INFORMATICS CENTRE

A Practice School-1 Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(JUNE,2020)

ACKNOWLEDGEMENT

We would like to express our gratitude to the Dean, Practice School Division, and his team, for making it feasible for us to take part in the Practice School-1 even during such difficult times. We would also like to thank our Practice school faculty, Professor Abhishek who has worked religiously in making all the preparations and arranging meets so that we do not face any problems.

Next, we would like to express our sincere vote of thanks to Jayanthi ma'am and Pechimuthu sir, for considering us for this internship and letting us work on the project: "STORING EDUCATIONAL CERTIFICATES USING BLOCKCHAIN". Their help and guidance have been an integral part of the working of the project.

We would also like to express our gratitude to our senior Rut Vora for helping us with different types of software.

We would also like to thank everyone else who was directly or indirectly involved to provide us with this opportunity.

The work summarized in this report represents the contribution of a group of students working on the project.

Tanya Prasad

Rajvi Sampat

Rohan Sinha

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)
Practice School Division

Station: **NATIONAL INFORMATICS CENTRE, NIC** Centre-Bengaluru
Bengaluru

Duration- 6 Weeks Start Date- 18/05/2020
Date of Submission- 25/06/2020
Title of the Project: STORING EDUCATIONAL CERTIFICATES USING BLOCKCHAIN

NAME	ID	DISCIPLINE
TANYA PRASAD	2018B5A71054	PHYSICS + COMPUTER SCIENCE
RAJVI SAMPAT	2018B4A70820G	MATH+COMPUTER SCIENCE
ROHAN SINHA	2018A7PS0417H	COMPUTER SCIENCE

EXPERTS:

Ms. Jayanthi S
Scientist-G, NIC

Mr. T Pechimuthu
Scientist-F, NIC

Name of PS Faculty: Mr Abhishek

Keywords: Blockchain, Smart Contract, Hyperledger Fabric, Nodes & Network
Project Areas: Blockchain Development

Abstract: Currently, at the time of admission to a university/school, a student has to submit a number of documents. After the submission, they are verified by the institute. There might be some
errors or false information in the docs provided. Sometimes the docs are referred back to the student. This whole system is quite cumbersome and time-consuming. We have to develop our own blockchain system on some platform in order to store the original documents digitally in it so that no tampering takes place. It also helps in avoiding the issue of duplication.

Date: 25/06/2020

Table of contents

1.Introduction:

1.1. About the National Informatics Centre(NIC)

The **National Informatics Centre** (**NIC**) is an attached office under the Ministry of Electronics and Information Technology (MeitY) in the Indian government. The NIC provides the infrastructure to help support the delivery of government IT services and the delivery of some of the initiatives of Digital India.

The National Informatics Centre (NIC) was established in 1976 under the Planning Commission by the India Government. In 1990, the takeoff by the Ministry of Electronics and Information Technology NIC's ICT Network facilitates the institutional linkages with the Ministries/ Departments of the Central Government, State Governments and District administrations of India.NIC is noted for being the primary constructor of e-Government applications.

NIC has a Center for Excellence in Blockchain Technology, which was the department under which we have done our project.

1.2. About the Project

Our project is to create a web application to store educational certificates, using a blockchain network.

Educational certificates issued to students will be stored by authorities directly on a blockchain network, which can be accessed by students from the website.

The blockchain network will serve as an immutable ledger of certificates, which can be updated only by authorities, and hence will prevent fraud of tampering and using fake certificates. The distributed and decentralized nature of the network ensures that it even more trusted, as no central organization can tamper with the marksheets.

Additionally, students can easily access this verified copy from any part of the world, in absence of the original hard copy. This is useful while applying for higher education or jobs, in order to be able to produce an authentic copy which can be trusted by the authorities.

1.3. What is Blockchain

1.3.1 Definition

Blockchain is a decentralised ledger which stores digital assets. Immutable data can be stored on the blockchain such that it is secured by cryptographic techniques and cannot be altered.

1.3.2 Block

The block consists of the transactions stored as an ordered set, and each block is linked to the previous block, as it contains the cryptographic hash function of the previous block.

1.3.3 Peer

Peer nodes are the basic network entities, each of which can hold copies of ledger and smart contracts. They maintain the ledger and depending on the type of peer node, can perform operations to the ledger. Peers are owned and maintained by members.

1.3.4 Ledger

The ledger is a channel's chain and current state data which is maintained by each peer on the channel.

1.3.5 Chaincode

Smart contracts are lines of code; domain specific programs, that define the transaction logic of an object in the blockchain. A chaincode is a technical container of a group of related smart contracts. The chaincode governs how smart contracts are packaged for deployments. Multiple smart contracts can be defined in the chaincode, and when the chaincode is deployed to the blockchain, all the smart contracts within it are made available. Chaincode enforces the rules for reading or altering key-value pairs or other state database information. Chaincode functions execute against the ledger's current state database and are initiated through a transaction proposal. Chaincode execution results in a set of key-value writes(write set) that can be submitted to the network and applied to the ledger on all peers.

1.3.6 Member

A blockchain consists of one or more members, which are unique and legally separate entities in the network. A member, such as an organization, will run one or more separate peer nodes. Network components will be linked to a member.

1.3.7 Ordering service

The ordering service consists of ordered nodes, which arrange the transactions in a chronological fashion. The ordering service exists independent of the peer processes, and the order transactions on first-come-first-serve basis for all the channels.The ordering service is common to the whole network, and contains the cryptographic identity material linked to each member.

1.3.8 Channel

A channel is a component of a private blockchain which allows for data isolation and confidentiality. A ledger can be made channel specific, and will be only shared across the peers in that channel. To interact with a channel, transacting parties must be properly authenticated to it.A channel is defined by members(organization), anchor peers per member, the shared ledger, chaincode applications and the ordering service nodes. The consortium is the list of organizations that are allowed to create channels.

1.3.9 Consensus

A consensus is a broader term overarching the whole network. The peer nodes come to an agreement about an order, which serves to maintain correctness of the transactions. Common consensus algorithms include proof of work and proof of stake.

1.3.10 Types of Peers

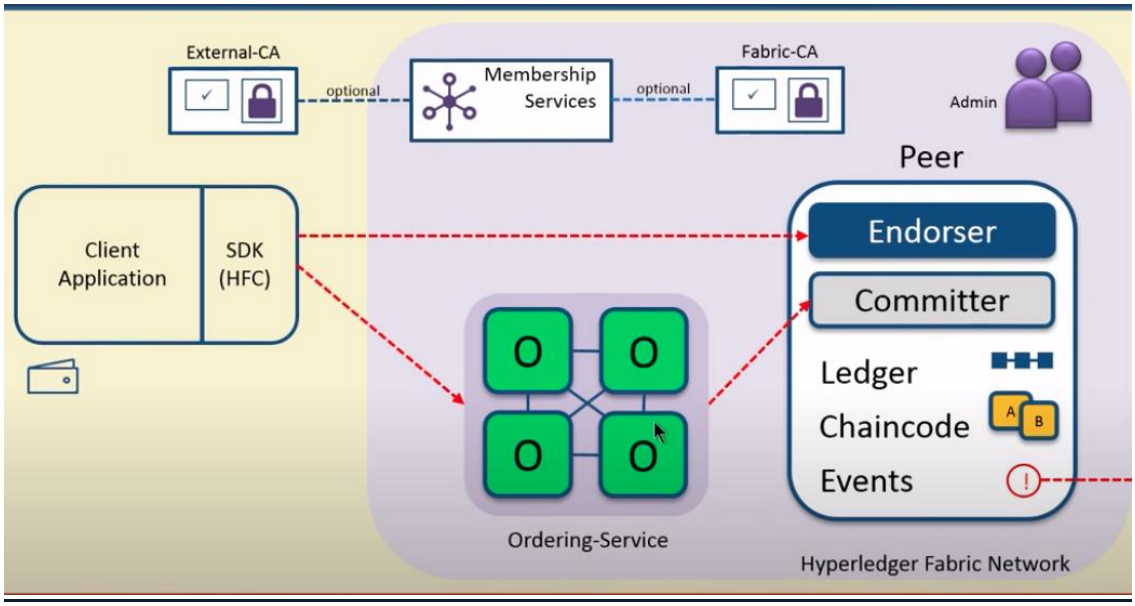
1. Committing Peer: Maintains the ledger and state, commits transactions and may hold smart contracts

2. Endorsing Peer: It is a specialized committing peer that receives a transaction proposal for endorsement, responds granting or denying endorsement. Must hold smart contract

3. Ordering node: Approves the inclusion of transaction blocks into the ledger and communicates with committing and endorsing peer nodes. Doesn't hold the smart contract and ledger.

1.3.11 Types of Blockchain Networks

Blockchain networks can be public, such as Bitcoin and Ethereum, where any node can write data on immutable ledger, or private, such as hyperledger and multichain, where only select nodes can write data on the ledger.



1.4 Hyperledger Fabric

Hyperledger fabric is an open-source, enterprise-graded permission distributed ledger technology platform. It is used majorly in enterprise contexts and has differentiating capabilities over popular distributed ledger or blockchain platforms.

Hyperledger CA:

The Hyperledger fabric CA is the default Certificate authority component. Which issues PKI-based certificates to network member organizations and their users. The CA issues one root certificate to each member and one enrollment certificate to each authorized user.

2.Project Description

Our project includes creating a Hyperledger Fabric blockchain network to store Marksheets, a website which from which students can view their marksheets and admins can add, modify and view marksheets, and an API which will read data and process requests and transfers from the website to the blockchain network. We have made another small application to create channels in the blockchain network.

2.1 Blockchain Network

2.1.1 Description

Our blockchain network will store the marksheets as a JSON object. We have used Hyperledger Fabric as our blockchain network. The marksheets will form an immutable ledger which cannot be tampered with. Since Fabric is a permissioned network, only the admins will have the ability to add marksheets. Through the API, a transaction request is sent to the endorsing peers. The endorsing peers will validate the transaction and will reach a consensus whether the transaction is valid. in case of viewing marksheets,i.e querying the ledger, the marksheets will be returned to the API in case of consensus. In case of adding or modifying marksheets, the application will check if the endorsement policy is fulfilled and submit the transaction to the Ordering service, which will arrange the transactions in chronological order and send the block of marksheets to all the leader peers in the organisation. The ordering service will have the list of leader peers through a channel, and the leader peers will deliver the block to all the peers in their organisation through gossip protocol, using the channel of the respective organisation. The ledger will then be updated, and the API will be notified once the marksheet is successfully added , and will update the website. The organisation will be all the centres responsible for updating the blockchain, such as the NIC centres, board etc. We have deployed the chaincode, which is a technical container of a group of related smart contracts, which will automate the transactions and ensure they are all following the same rules. Our smart contract stores the marksheet data of each student as a JSON object.

2.1.2 Tools used

1. Hyperledger Fabric

2. Docker
3. GoLang to write the chaincode

2.1.3 Working

We changed the chaincode according to the requirements of our certificate and then deployed it into the network. Marksheet data will be stored as a JSON object. Our network currently consists of 2 organisations with 1 peer each and 1 ordering service with 1 peer. We linked the blockchain to the API which will be able to request transactions.

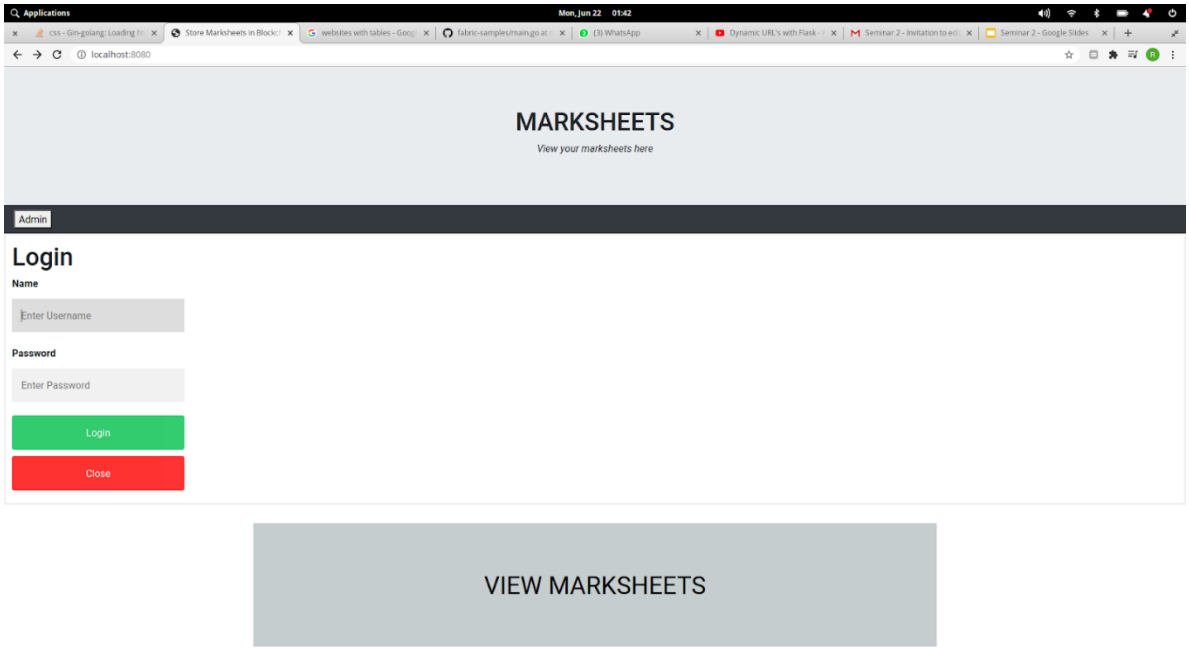
2.2 Website

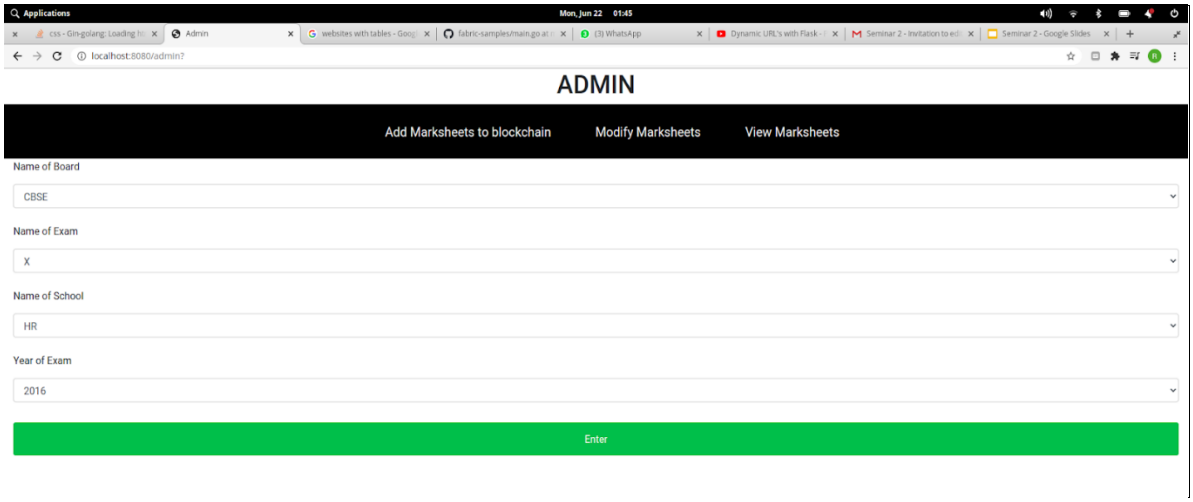
2.2.1 Description

Our front end website consists of a home page, from which students can view their marksheets and admins can login. The students have to enter their credentials: Board, Name of Examination, Year of Examination and Roll number in a form, which will send a request to the API and retrieve the marksheet from the blockchain and display it to the student. The admins have to enter their login credentials, after which they will be able to add, modify or view marksheets. On entering the board,year, school and exam, they will receive a list of students in that school with the respective roll number, and once the admin uploads the marksheet, the name will be flagged. The data of the boards and schools will be stored in an SQLite relational database and the form is populated using the same. The database is queried by the API to give the school wise data

2.2.2 Tools used

We have used HTML, CSS and Javascript to design the website. The API endpoints are called using HTTP get and post requests, using AJAX and JQuery.





2.2.3 Working

We have made the website with the necessary forms. We have linked the forms to the database and to the API, so that the data can be retrieved from the database and the marksheets can be added to the blockchain and viewed on request.

A comfortable and easy-to-use User Interface has been prepared.

2.3 API

2.3.1 Description

The application programming interface consists of 3 end points, the **RDBMS**, which will consist of the board, school and student data, the **Blockchain network**, which will have the marksheets stored as JSON objects, and the **website**.

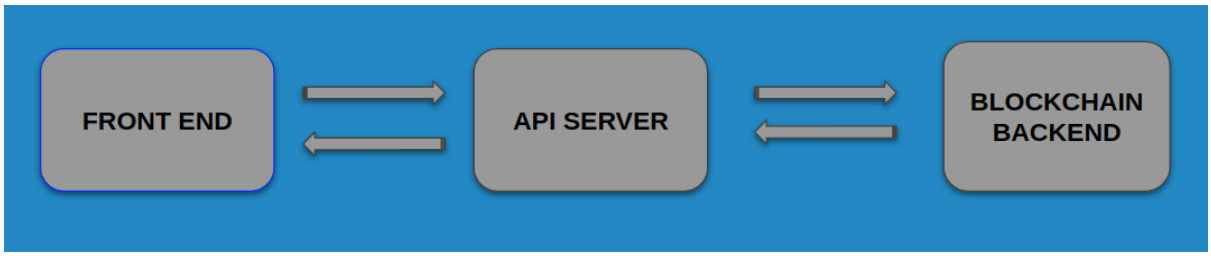
2.3.2 Tools used

The API has been written in GO lang, using the Gin framework. GORM, which is a very powerful Object Relational Mapping library for Golang, has also been used.

2.3.3 Working

We firstly learnt the basics of GOLang, understood how the API will function, and learnt the basics of the Gin framework. After deciding the endpoints etc., we prepared it. HTTP requests are being sent and received using ajax requests.

The data for the dropdown forms can be fetched from the database using the API, the values entered will be sent back to the API on submission which will accordingly fetch the marksheets from the blockchain, or add new marksheets to the blockchain as JSON.



2.3.4 API Endpoints

1. r.GET("/student.html", student)
2. r.GET("/", home)
3. r.GET("/admin",admin)
4. r.POST("/add",add_)
5. r.POST("/modify", mod)
6. r.POST("/view",view)
7. r.POST("/add_record",add)
8. r.POST("modify_record",modify)
9. r.GET("/find/",find)

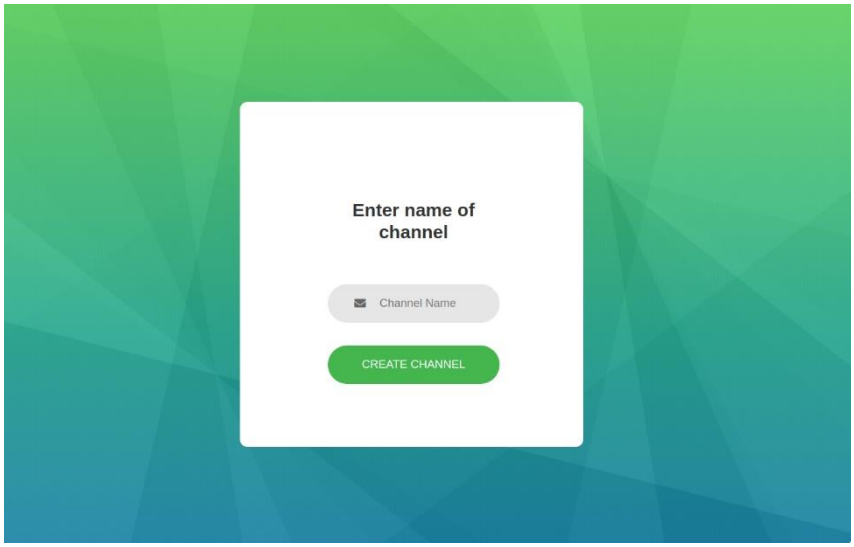
2.3.5 Structure of our certificate

```
type Student struct {
    Name  string `json:"name"`
    Year  string `json:"year"`
    Board string `json:"board"`
    Mark  string `json:"mark"`
    Roll  string `json:"rollno"`
}
```

2.4 Admin Access

Apart from all the above milestones, we have also prepared an application for the Admin. It authorizes the Admin to add nodes, turning up the network and adding channels to the blockchain. Upon deployment at a national level, more functionalities can be added into it.

A snapshot of the Admin's Frontend:



3.Conclusion

We have understood and learnt about blockchain technology and how it can be used along with a web application. We have learnt various skills, such as using Hyperledger fabric, docker, GO lang, Gin framework, HTML, CSS and Javascript, SQL in the course of this project. We have learnt about the importance and benefits of blockchain technology and hope to implement it in the future as well in various of its other use cases.

3.1 Major Milestones

1. Making the front end of the website with functional forms and displaying or adding data from the blockchain
2. Preparing the chaincode for blockchain and deploying the blockchain network.
3. Preparing RDBMS for Boards and Marksheet details.
4. Building the API to interact with both the frontend and the backend blockchain, by calling necessary functions at the endpoints
5. Writing HTTP requests to the server.

3.2 Technical Skills Acquired

1. We learnt about blockchain technology, it's advantages, use cases, architecture and technical aspects. We learnt how to implement it in a web application.
2. We learnt how to create channels and deploy the blockchain network using Hyperledger Fabric and wrote our own chaincode.
3. Learnt how to run UP and bring DOWN our established network using CRYPTOGEN TOOL. It is also used to generate Certificates and keys.
4. Using Docker.
5. We learnt Go lang and the Gin framework.
6. We learnt how APIs and learnt how to make our own API in Go.
7. We learnt web development and gained experience in using HTML,CSS and Javascript.We learnt how to use Ajax and make HTTP requests to link the API to the frontend and backend.
8. We learned how to use Git and GitHub to collaborate amongst ourselves.

3.3 Soft Skills Learnt

1. Team Work. At numerous points of the project we divided the work among ourselves and thus were able to prepare everything on time. Inspite of the remote nature of the project, we had regular meetings and learned how to coordinate with team members. We made the most of each other's strengths and helped each other along the way. We also had to understand the work done and code written by each other and finally collaborate and merge it into a single project.
2. Communication skills. We gained experience by participating in a Group Discussion. We learned how to present our points logically and effectively, while also listening to and understanding other people's points
3. Presentation skills: We also gave seminars presenting our work, where we learned to make good presentations and explain our work concisely and effectively.
4. Confidence; We gained confidence through our various meetings, seminars and the group discussion.

4.References:

1. NIC Centre of Excellence in Blockchain Technology website: https://blockchain.gov.in/
2. NIC official Website: https://www.nic.in/
3. Hyperledger Fabric Official Documentation: https://hyperledger-fabric.readthedocs.io/en/latest/getting_started.html
4. GoLang Official Documentation: https://golang.org/doc/
5. NPTEL blockchain: https://www.youtube.com/channel/UCYa1WtI-vb_bx-anHdmpNfA
6. CS 50 web programming course: https://cs50.harvard.edu/web/2018/
7. EdX Blockchain course: https://courses.edx.org/courses/course-v1:LinuxFoundationX+LFS170x+1T2020/
8. Gin framework: https://gin-gonic.com https://github.com/gin-gonic