

**ELEC S411F (2020/21)**

**Electronic and Computer Engineering Project**

**Final Report**

|  |  |
| --- | --- |
| **Blockchain Research and Implementation(S07)** | |
|  |  |
| Project number: | S07 |
| Student name | Yung Ho Chun (12424647) |
| Supervisor: | Dr. Steven Choy |
| Submission Date: | 13/6/2020 |

**Declaration of Originality**

I, Yung Ho Chun , declare that this report and the work reported herein was composed by and originated entirely from me. This report has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the reference section.

6/3/2021

# Abstract

This project describes the design and development of the storage of university certificates and graduation certificates. The system needs to correctly record the student's information, ensure the security of the information, and correctly display the student's information. An important part of the project is to record the data while ensuring that the data cannot be easily modified by others to ensure the credibility of the data.

Through the connection between the website and the blockchain in the project, you can store data for a long time without relying on server records. At the same time, it can also ensure that the information will not be easily modified by others. In addition, the reliability of university certificate information can be ensured through the blockchain. The blockchain ensures that the student is a graduate of the Open University by recording the year of graduation, student number, student name in Chinese and English, and student's Chinese and English course name.

The system I made can record the graduation certificate information of a large number of university students, and at the same time ensure that users are easy to use this website, and it is easier to prove the accuracy and reliability of the graduation certificate information to the employer.

Of course, there are also imperfections in the project, which can still be modified and improved in future work. Although it uses the decentralization of the blockchain and cannot be modified at will, it does not fully utilize the encryption of the blockchain, so the technology of the blockchain can be used again. Although the encryption feature is not necessary in this system, the encryption feature can add more security to the system and make users feel more at ease. So in the future, we will improve this function so that more users can use this system.

Table of Contents

[1. Abstract 2](#_Toc74512966)

[2. Introduction 8](#_Toc74512967)

[2.1 Project Objectives 8](#_Toc74512968)

[2.2 Organization of the Report 8](#_Toc74512969)

[3. Background 9](#_Toc74512970)

[3.1 What is blockchain 9](#_Toc74512971)

[3.2 Current state of blockchain technology 10](#_Toc74512972)

[3.3 The Comparison of Blockchain and Database 11](#_Toc74512973)

[3.3.1 authority 11](#_Toc74512974)

[3.3.2 architecture 11](#_Toc74512975)

[3.3.3 data processing 12](#_Toc74512976)

[3.3.4 Data integrity 12](#_Toc74512977)

[3.3.5 transparency 13](#_Toc74512978)

[3.3.6 cost 13](#_Toc74512979)

[3.3.7 which performed 14](#_Toc74512980)

[3.4 What is Ethereum 15](#_Toc74512981)

[3.5 what is hyperledger 17](#_Toc74512982)

[3.6 What is Hyperledger Fabric 18](#_Toc74512983)

[3.7 what is Consensus model 19](#_Toc74512984)

[3.7.1 PoW 19](#_Toc74512985)

[3.8 Comparison of blockchain technology solutions 21](#_Toc74512986)

[3.9 smart contract 23](#_Toc74512987)

[3.10 DApp 24](#_Toc74512988)

[3.11 What is asp 26](#_Toc74512989)

[3.12 What is php 27](#_Toc74512990)

[3.13 What is jsp 28](#_Toc74512991)

[3.14 Comparison of Dynamic web design language 29](#_Toc74512992)

[4. Design 30](#_Toc74512993)

[4.1 System architecture 30](#_Toc74512994)

[4.2 Functional Design 31](#_Toc74512995)

[4.3 Database Design 32](#_Toc74512996)

[4.4 Blockchain Design 33](#_Toc74512997)

[4.5 Web Application Design 34](#_Toc74512998)

[4.5.1 Login 35](#_Toc74512999)

[4.5.2 create student date 38](#_Toc74513000)

[4.5.3 show student data 38](#_Toc74513001)

[4.5.4 create admin 38](#_Toc74513002)

[4.5.5 change student data 38](#_Toc74513003)

[4.6 Backend Design 39](#_Toc74513004)

[5. Experiment 40](#_Toc74513005)

[5.1 Create blockchain server 40](#_Toc74513006)

[5.1.1 install git 40](#_Toc74513007)

[5.1.2 Check git version 42](#_Toc74513008)

[5.1.3 Install geth 42](#_Toc74513009)

[5.1.4 check geth 45](#_Toc74513010)

[5.2 build a private blockchain 46](#_Toc74513011)

[5.2.1 Create a folder to store private blockchain data files 46](#_Toc74513012)

[5.2.2 generate genesis.json file 46](#_Toc74513013)

[5.2.3 create a genesis block 52](#_Toc74513014)

[5.3 testing blockchain 55](#_Toc74513015)

[5.4 Smart contract 57](#_Toc74513016)

[6. Results and Discussion 60](#_Toc74513017)

[6.1 Login 60](#_Toc74513018)

[6.2 create and show student date 62](#_Toc74513019)

[6.3 create admin 65](#_Toc74513020)

[6.4 change student date 67](#_Toc74513021)

[7. Conclusion 69](#_Toc74513022)

[8. References 70](#_Toc74513023)

[9. Appendix 72](#_Toc74513024)

[9.1 remix code(Storage.sol) 72](#_Toc74513025)

List of FIGURES

[Figure 2.1 Hyperledger Fabric architecture 18](#_Toc74513065)

[Figure 2.2 Illustration of a DApp that uses a blockchain with smart contracts combined with the pillars of Swarm 25](#_Toc74513066)

[Figure 3.1 Use Case Diagram 31](#_Toc74513067)

[Figure 3.2 Entity Relationship Diagram 32](#_Toc74513068)

[Figure 3.3 the structure of the entire website 34](#_Toc74513069)

[Figure 3.4 Sequence diagram for login 36](#_Toc74513070)

[Figure 4.1 command#1 result 40](#_Toc74513071)

[Figure 4.2 command#2 result 41](#_Toc74513072)

[Figure 4.3 command#3 result 41](#_Toc74513073)

[Figure 4.4 command#4 result 42](#_Toc74513074)

[Figure 4.5 command#5 result 43](#_Toc74513075)

[Figure 4.6 command#6 result 43](#_Toc74513076)

[Figure 4.7 command#7 result 43](#_Toc74513077)

[Figure 4.8 command#8 result 44](#_Toc74513078)

[Figure 4.9 command#9 result 45](#_Toc74513079)

[Figure 4.10 command#13 result 50](#_Toc74513080)

[Figure 4.11 private\_chain folder 51](#_Toc74513081)

[Figure 4.12 command#15 result 52](#_Toc74513082)

[Figure 4.13 data folder 53](#_Toc74513083)

[Figure 4.14 command#16 result 54](#_Toc74513084)

[Figure 4.15 command#17 result 56](#_Toc74513085)

[Figure 4.16 command#18-21 result 56](#_Toc74513086)

[Figure 4.17 command#22 result 56](#_Toc74513087)

[Figure 4.18 command#23 result 56](#_Toc74513088)

[Figure 4.19 blockchain coding 57](#_Toc74513089)

[Figure 4.20 building coding 58](#_Toc74513090)

[Figure 4.21 upload coding to blockchain 58](#_Toc74513091)

[Figure 4.22 upload coding successful to blockchain 59](#_Toc74513092)

[Figure 5.1 login page 60](#_Toc74513093)

[Figure 5.2 welcome(admin).html 61](#_Toc74513094)

[Figure 5.3 enter student information 63](#_Toc74513095)

[Figure 5.4 upload data in blockchain 63](#_Toc74513096)

[Figure 5.5 database date 64](#_Toc74513097)

[Figure 5.6 display student graduation information 64](#_Toc74513098)

[Figure 5.7 create admin data 65](#_Toc74513099)

[Figure 5.8 create account successful 66](#_Toc74513100)

[Figure 5.9 change student information successful 67](#_Toc74513101)

[Figure 5.10 upload data in blockchain 67](#_Toc74513102)

[Figure 5.11 display student graduation information 68](#_Toc74513103)

List of TABLES

[Table 2.1 comparison of blockchain technology solutions 21](#_Toc74513116)

[Table 3.1 Blockchain Design Overview 33](#_Toc74513117)

[Table 3.2 Backend Files Overview 39](#_Toc74513118)

[Table 3.3 Backend Functions Overview 39](#_Toc74513119)

[Table 4.1 Install git command 40](#_Toc74513120)

[Table 4.2 Check git version command 42](#_Toc74513121)

[Table 4.3 Install geth command 42](#_Toc74513122)

[Table 4.4 check geth command 45](#_Toc74513123)

[Table 4.5 Create file command 46](#_Toc74513124)

[Table 4.6 generate genesis.json file command 49](#_Toc74513125)

[Table 4.7 create a genesis block command 52](#_Toc74513126)

[Table 4.8 testing blockchain command 55](#_Toc74513127)

# Introduction

Today, social work is becoming more and more professional, leading to higher and higher education requirements for work. However, many families cannot afford college tuition, which makes it difficult to go to college. Because of the above reasons, criminals in various places have begun to sell counterfeit certificates[1]. Although there are some anti-counterfeiting measures on genuine certificates, criminals have begun to break through and produce counterfeit certificates that cannot be distinguished by professionals. Therefore, in order to solve this problem, new technology can be used to replace the physical certificate, making it difficult for criminals to break. Among them, distributed applications can be used to store graduation certificates. Therefore, an OUHK decentralized application is in need.

## Project Objectives

The project is to design and develop a decentralized application for the students of the Open University of Hong Kong. This decentralized application will run on different clients. It will provide storage and display of graduation certificate information, such as name, graduation date, class honours, degree, university name and other information.

There are three goals to be achieved.

* Establish the Ethereum, store the data needed for each person's graduation certificate on each node, making it difficult to change the data.
* Establish a smart contract, which can connect the data entered at the front desk with the Ethereum.
* To establish the front desk, we can use html to create all the things needed in the front desk.

## Organization of the Report

Chapter 3 introduces a background of Blockchain and related technologies.

Chapter 4 presents an overview of the frontend and backend design of website.

Chapter 5 describes the implementation details and the deliverables for the whole system.

Chapter 6 concludes the overall of the project and identifies some further work.

# Background

This chapter includes 7 parts. Section 2.1 shows what a blockchain is. Section 2.2 gives the current state of blockchain technology, and Section 2.3 gives the meaning of Ethereum. Related blockchain technology solutions will be compared in Section 2.4. Section 2.5 shows SMART CONTRACT. Finally, DAPP technology will be described in section 2.6.

## What is blockchain

Blockchain[2][3][4][5][6] was originally a special database technology designed for Bitcoin by a person with the pseudonym Satoshi Nakamoto. From a data point of view, blockchain is a data structure that combines blocks in a chain, which enables participants to establish a consensus on the sequence of events and current status of transactions recorded on the entire network. In a nutshell, blockchain refers to the technology of collectively maintaining a reliable database through decentralization and trust lessness. The key technical points involved in the blockchain include decentralization, trust lessness, collective maintenance, reliable databases, timestamps, asymmetric encryption, etc. Blockchain redefines the way credit is generated in the network. In the system, participants do not need to know the background information of other people, nor do they need to rely on third-party guarantees or guarantees. Blockchain guarantees the system's activities for value transfer for recording, transmission, and storage, the final result must be credible.

## Current state of blockchain technology

After Bitcoin, diversified blockchains have developed together. After nearly 8 years of development and practice, people have a deeper and deeper understanding of blockchain technology, and they are also in the process of making blockchain technology more compatible with their own needs. Gradually understood the following two points:

Blockchain technology is essentially the underlying technology of the Internet protocol and the technology of the Internet data level.

The "decentralization" advocated by the blockchain is not an anti-center, but a sub-center (the most thorough sub-center is that each node becomes its own center).

Therefore, many fields try to make further improvements based on the Bitcoin blockchain. At present, the blockchain has developed from a completely decentralized public blockchain in Bitcoin to a side chain attached to the public chain and a private blockchain that is not completely decentralized. Blockchain technology is at the theoretical stage and needs to be practiced. Blockchain technology emerged with the birth of Bitcoin, and the most mature blockchain currently is Bitcoin's blockchain. The input and use of blockchain technology at home and abroad has been gradually expanded, but there is no fully implemented application result yet. From a development point of view, blockchain technology is still in the theoretical stage, and it will take a long time to practice in the future for technology conversion.

## The Comparison of Blockchain and Database

I think blockchain is the best because of the following reasons [16].

### authority

The first major difference between relational databases and blockchain technology is that they have different authoritative systems. In relational database systems, there is always a form of centralized authority.

There is no decentralization of any kind in the architecture model of the system. Basically, its role is to provide the administrative authority with sole control, and they can make changes as they wish.

On the other hand, blockchain technology provides decentralized authority. This means that it does not have any central authority or middleman in maintaining the ledger.

Therefore, the user has complete control over what will happen in the system. Therefore, there is no central agency that can make changes at will.

### architecture

Another huge difference between relational databases and blockchain technology is that they have different architectures in terms of ledger systems. Basically, relational database systems have a typical client-server model. Although it has been the main structure of our Internet system for a long time, there are certain limits in this respect.

In fact, the client-server model is very susceptible to hacker attacks, and is attacked by hackers from time to time. On the other hand, blockchain provides a peer-to-peer architecture instead of a client-server architecture. Here, users on the node can connect to each other using encryption protocols. More importantly, it improves the security status of the ledger system, so it is less vulnerable to hacker attacks.

### data processing

The two technologies deal with data in very different ways. For relational databases, it supports CRUD. This means that in the system, users can create, read, update, and delete. In fact, with so much change or freedom to change, the value in the system can cause a lot of corruption. People can change the value to get more benefits.

More importantly, in many cases, management agencies can change or delete information that is vital to the company. It's all for my own self-interest.

On the other hand, the blockchain only gives you read and write permissions. More importantly, in many cases, it can also restrict public access to these two types of access. So, here, you can only insert data once, and then you can't update or delete it.

### Data integrity

Compared with relational database systems, blockchain has higher integrity. Anyone who tries to change the value in the transaction will obviously also change the hash ID.

Therefore, the block will be separated from the original chain and become invalid. More importantly, the blockchain provides other security layers at each vulnerability, such as additional security protocols in the authorization process. Therefore, technology maintains data integrity in every possible way.

But not in a relational database. In fact, they provide automatic error detection and forced automatic filling. This means that no one can leave any row or column in the table blank. More importantly, it can also outline what kind of information will appear there, such as numbers or characters. But it cannot prevent others from changing the information.

### transparency

Another important feature of this technology is that it provides complete transparency when verifying data. Therefore, anyone on the network will trust the system because it operates based on a fully-proven algorithm.

More importantly, in the public blockchain, the ledger system is available for everyone to see. However, in private situations, viewing options may be restricted based on predefined criteria.

On the other hand, relational database systems provide no transparency at all. It is completely centralized, and users cannot know whether the database has correct information.

More importantly, they cannot even verify whether these are right or wrong. As a result, users are slowly losing trust in the system.

### cost

Well, I think this round is technically a relational database. Relational databases are legacy networks, and they have been around for a long time.

Therefore, implementing them is not time-consuming. However, compared with traditional databases, relational databases do require more time to set up. But it is also very cheap.

On the other hand, the blockchain is now just a new member of the world. It is also on the verge of continuous development, so it is very complicated to set up. Therefore, the resources to implement the blockchain are a bit expensive. Even the talents you hire for work have higher pay.

### which performed

This attribute provides a more mixed output. In fact, when it comes to relational databases, you will get faster output compared to blockchain. Since the relational database system does not have complicated functions, such as consensus or any other agreement that may slow down the system.

More importantly, when the central agency manages the system, there will be no overwhelming traffic occupying all bandwidth.

On the other hand, when the number of nodes is limited, the blockchain is much faster. But when this number starts to rise, the system will slow down over time. Therefore, the performance of the blockchain changes over time.

## What is Ethereum

Ethereum[7][8] is a digital currency and blockchain platform focusing on the application of smart contracts. Its decentralized technology frees transactions from the hassle of downtime and review, resolves thorny issues such as traditional contract disputes, and can also avoid fraud and Interference from a third party. The goal of Ethereum is to provide a blockchain with a built-in mature Turing complete language. With this language, contracts can be created to encode arbitrary state transition functions. Users can simply implement logic with a few lines of code. Create a variety of systems that meet the needs.

From the perspective of technical architecture, Ethereum is a universal distributed application development platform and programming language that has nothing to do with the underlying blockchain and protocol, including digital currency Ether (Ether) and EtherScript (EtherScript), used to build and release distribution Application. It has the characteristics of openness and generality, and has a built-in Turing complete virtual machine, which can use any currency, protocol and blockchain. Every node on the network can run the Ethereum virtual machine to publish distributed smart contract programs.

Ethereum has its own distributed system: including file service Swarm, information transmission Whisper and reputation guarantee. Swarm is a decentralized file service; Whisper is an encrypted communication transmission system; credit guarantee provides a trustless network to establish reputation and reduce discovery systems, which can be provided by a third party.

In addition, Ethereum itself is also open-source software, with the rapid innovation characteristics of open-source projects. Through code sharing, new projects can quickly iterate to products that meet market needs. At present, Ethereum is being closely watched by some financial institutions, bank consortia, and large companies such as Samsung, Deloitte, RWE, and IBM. This has also spawned several simplified and automated financial transactions, Blockchain applications such as merchant loyalty index tracking, gift cards aimed at decentralization of electronic transactions, etc.

Ethereum has received strong support from the technical community and commercial institutions in the past two years and has developed rapidly. It is likely to become one of the most competitive solutions in the blockchain field in the future.

## what is hyperledger

Hyperledger[13] is an open source project initiated by the Linux Foundation in 2015 to promote blockchain digital technology and transaction verification. It is composed of 30 founding company members and a set of technical and organizational governance institutions.

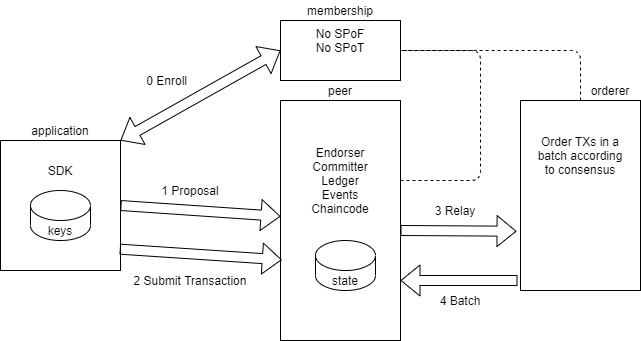
The origin of the Linux Foundation: Founded in 2000, it was derived from the concept of community sharing. At that time, in order to build a software ecosystem, the foundation was used to manage open-source projects, and such a community development code was established so that everyone could know This code is safe and usable. Everyone is easy to obtain because the code is open source. Everyone can also benefit from it. Such a foundation has been running for the eighteenth year.

The purpose of Hyperledger-to build a blockchain framework and platform by the software developer community.

* Create an enterprise-level, open source, distributed ledger framework and code base to support commercial transactions.
* Provide a neutral, open and community-driven infrastructure supported by technology and business governance.
* Educate the public about the market opportunities of blockchain technology.
* Promote our community’s toolkit approach to many platforms and frameworks.

## What is Hyperledger Fabric

Hyperledger Fabric[14][15] is an enterprise blockchain framework. It mainly provides a development foundation for blockchain applications and solutions that adopt a modular architecture, pursuing Modular, Scalabl, and Secure. Hyperledger Fabric can make Consensu, Membership Services, etc. Plug-and-Play plug-and-play. The smart contract in Hyperledger Fabric can be implemented through Chaincode.



#### Figure 2.1 Hyperledger Fabric architecture

In Figure 2.1, it show that Hyperledger Fabric decomposes Peer nodes into Endorser, Committer, Ledger, Events, and Chaincode according to their functions. At the same time, it extracts the consensus mechanism from the nodes and independently becomes an Orderer node to provide a plug-and-play consensus mechanism.

## what is Consensus model

The way blockchain works is often confusing. Because people have doubts about how this system will replace banks. Banks get high commissions from their transactions, and these commission rates generally range from 10% to 20%.

However, using blockchain technology, this commission rate is almost one in a thousand. When we exchange cryptocurrencies on the blockchain, we need someone to help with these transactions. In this way, we pay commissions to witnesses who conduct these transactions.

The difference between a bank and a witness is the speed and reliability of transactions, and the difference between the commissions they receive. Compared with banks, blockchain is faster, more reliable, and cheaper.

### PoW

The establishment of a proof-of-work agreement to successfully reach a consensus between decentralized devices can be said to be the highest achievement of Bitcoin founder Satoshi Nakamoto. In this process, he laid the foundation for the revolutionary technology of blockchain.

Proof of Work (PoW) is a consensus protocol introduced by Pioneer and Bitcoin, and is widely used in many other blockchain projects. This process is usually called "mining", so the nodes on the network are called "miners". PoW appears in the form of answers to mathematical questions. This question requires a lot of work to get, but once the answer is obtained, it is easy to verify that it is correct.

Looking back, let's talk about "nodes". "A node is a powerful computer that executes Bitcoin software. It helps Bitcoin keep executing by participating in information transmission. Anyone can run a node. You only need to download the Bitcoin software (free) and keep an open port (the disadvantage is that it It consumes energy and storage space-at the time of writing this article, the network occupies about 145GB). Nodes spread bitcoin transactions on the network. A node will send information to several nodes it knows, and then who forwards the information to them knows Node, etc. In this way, it can quickly spread throughout the network.

Some nodes are mining nodes (usually called "miners"). These group the outstanding transactions into blocks and add them to the blockchain. When it is combined with the data in the block and passed through the hash value function, it will produce a result within a certain range. This is more difficult than it sounds.

This process includes ensuring that each confirmed block in the chain rewards the miner in cryptocurrency, and the transaction fee charged by the miner through the network to transmit the currency, as well as any predetermined rewards, to mine these cryptocurrencies. It ensures that miners are incentivized and continue to maintain the blockchain because they are rewarded for doing so.

These rewards are particularly important because the puzzle process being solved is very expensive and needs to be done in terms of time and computing power. Maintaining the enthusiasm of these miners is a key function of the protocol, because they are, in a sense, the basis for maintaining the execution of the system. The use of proof-of-work systems is to prevent counterfeiting transactions, because it is extremely difficult to counterfeit the information required for transactions, but it is easy to verify.

## Comparison of blockchain technology solutions

|  |  |  |
| --- | --- | --- |
|  | ETHEREUM | HYPERLEDGER FABRIC |
| Need cryptocurrency | Ethereum or User-created | No |
| internet | Public chain or permission chain | Need permission |
| transaction | Anonymous or private | Public or confidential |
| consensus | PoW | PBFT |
| Smart contract | Yes | Yes |
| Language | Golang, C++, Python | Golang, Java |

##### Table 2.1 comparison of blockchain technology solutions

In table 2.1, Both Hyperledger and Ethereum are very flexible, but their advantages lie in different aspects. Ethereum's powerful smart contract engine makes it a universal platform for any application.

Also, the Hyperledger Fabric project provides users with many consensus mechanisms that can be used. It uses the Kafka consensus algorithm natively. It also mixes the Solo and Raft protocols. The Solo protocol is very suitable for developers, and Raft is crash tolerant.

The Ethereum consensus mechanism uses an upgraded version of the proof-of-work algorithm. Although the Ethereum consensus mechanism is very efficient and robust, it requires a lot of resources and slows down. However, the Ethereum consensus mechanism PoW is superior to the Raft protocol.

Ethereum is a public blockchain platform, so you don't need to gain access. Anyone can download the Ethereum framework to participate in Ethereum mining and discover and access the transactions that occur in it.

On the other hand, Hyperledger maintains strict control over accessibility. Only authorized members can access the platform and use its tools. In addition, each participant needs to obtain permission to join the Hyperledger network.

After considering all the above factors, because Ethereum does not require access rights and consensus mechanism, I think Ethereum is more suitable for this project.

## smart contract

A smart contract[9][10][11][12] is a special agreement used when formulating a contract in the blockchain. It is mainly used to provide verification and execution of the conditions set in the smart contract. The smart contract contains code functions, and can also interact with other contracts, make decisions, store data, and send Ether.

What is more special is that these transactions are traceable, difficult to tamper with, and irreversible, which enables smart contracts to conduct secure transactions without a third party. In addition, the smart contract is defined by the creator and constructed by the execution of the blockchain network. All the information related to the contract terms in the smart contract is automatically executed in accordance with the operations set in the contract.

Then, there are three main elements to create a smart contract: Subject of Contract, digital signature, contract terms and decentralized platform.

Subject of Contract: There must be a contract subject in the smart contract, so that the related goods and services in the contract can be automatically locked and unlocked in the smart contract program.

Digital signature: Smart contracts require all participants to authenticate with their private key before it can be activated.

Contract terms: All the operating sequences of the terms in the smart contract must be approved and signed by all participants before they can be executed.

Decentralized platform: Smart contracts are put into a decentralized blockchain platform and distributed among various nodes, waiting for the contract to be executed.

## DApp

Decentralized Application is an application developed based on the core of blockchain technology.

According to The General Theory of Decentralized Applications White Paper, applications that meet the following criteria are considered decentralized applications:

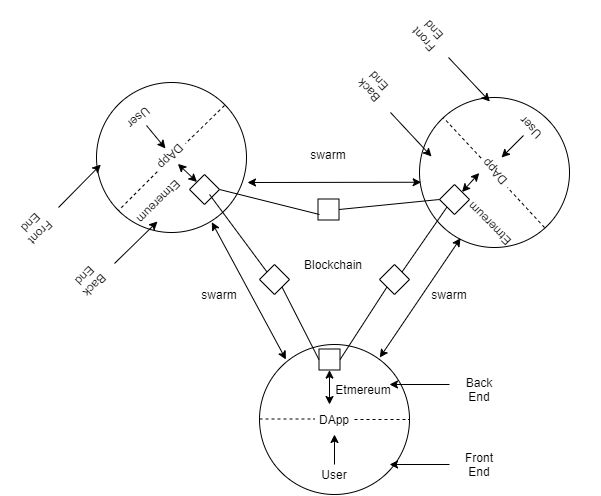
Open Source-Distributed applications must be completely open source and run autonomously. Simply put, the code is completely open for review and modification by other developers or programmers.

Decentralization-All data is encrypted and stored in a publicly accessible blockchain.

Tokens-provide Coins or Tokens for the blockchain to provide value for miners and users.

Algorithm-Use standard encryption algorithms to enable mining through a network of nodes.

In figure 2.2, it shows the structure of the entire for DApp.



#### Figure 2.2 Illustration of a DApp that uses a blockchain with smart contracts combined with the pillars of Swarm

## What is asp

ASP adopts the scripting language VBScript (JavaScript) as its development language. ASP is a dynamic web language developed by Microsoft, and it also inherits the tradition of Microsoft products. It can only be executed on Microsoft's server product IIS (Internet Information Server).

ASP is a development environment on the Web server side, which can generate and execute dynamic, interactive, and effective Web service applications. Its technical characteristics mainly include the following aspects:

* Browser Independence, the client only needs to use a browser that can execute HTML code to browse the web content designed by Active Server Pages. The scripting languages ​​(VBScript, JScript) used by Active Server Pages are all executed on the Web server side, and the client browser does not need to execute these scripting languages.
* Active Server Pages can be compatible with any ActiveX Scripting language. In addition to VBScript or JScript language design, you can also use other scripting languages ​​provided by third parties, such as REXX, Perl, Tel, etc., through plug-in. The script engine is a COM (Component Object Model) object that processes script programs.
* Using simple and easy-to-understand scripting languages ​​such as VBScript and JScript, combined with HTML code, you can quickly write website applications. You can use server-side scripts to generate client-side scripts.
* Use a common text editor, such as Windows Notepad, to design the program, without compiling, easy to write, and can be executed directly on the server side.

## What is php

PHP is a cross-platform server-side embedded scripting language. It almost always borrows the syntax of C, Java and Perl language, combined with PHP's own characteristics, so that Web developers can quickly write dynamic pages. The characteristic of PHP is that it supports most databases, and its source code is completely open.

PHP can be executed normally on Windows, Unix, and Linux web servers, and it also supports general web servers such as IIS and Apache. When users change platforms, there is no need to change the PHP code.

PHP and MySQL are currently an excellent combination. Users can also write their own peripheral functions to access the database indirectly. In this way, when changing the database, they can easily modify the code to adapt to such changes.

prompt:

PHP LIB is the most commonly used series of base libraries that can provide general business needs. However, the database interface support provided by PHP is not uniform with each other.

## What is jsp

JSP is similar to PHP and can be executed on almost all platforms, such as Windows, Linux, and Unix. Web server Apache has been able to support JSP, and Apache is widely used on Windows, Unix and Linux, so JSP has a wider execution platform.

Although the Windows operating system now occupies a large market share, Unix still has a great advantage in terms of servers, and the newly emerging Linux is even more powerful.

When porting from one platform to another, JSP and JavaBean do not even need to be recompiled, because Java bytecodes are all standard and platform-independent. ASP, PHP, and JSP are all technologies for Web servers, and the client browser does not require any additional software support.

Ordinary HTML pages only rely on the Web server, but ASP, PHP, and JSP pages require additional language engines to analyze and execute program codes. The execution result of the program code is re-embedded in the HTML code, and then sent to the browser together. All three provide the ability to mix some kind of program code in HTML code, and the language engine interprets and executes the program code. The JSP code is compiled into a Servlet and interpreted and executed by the Java virtual machine. This compilation operation only occurs when the JSP page is requested for the first time.

## Comparison of Dynamic web design language

After comparing three dynamic web design languages, I think PHP is more in line with the requirements of this project because of the following three points:

1. A development language that can learn quickly, cross-platform, and has good database interaction capabilities. What ASP can't compare to it is this kind of cross-platform ability, and it is this kind of ability that makes Unix/Linux have a development language comparable to ASP. The grammar is simple and easy to write. There are also a large number of books on the market, and there are also a large number of codes on the Internet that can be shared.

Enjoy, for a beginner who wants to learn some "advanced Unix" development is a decisive starting point.

2. It is closely integrated with Apache and other extension libraries. PHP and Apache can be combined in a statically compiled manner, and other extension libraries can also be combined in this way (except for the Windows platform). The biggest advantage of this approach is to maximize the use of cpu time and memory, while extremely effective use of the high-performance throughput of Apache

ability. At the same time, the external expansion is also statically built, so as to achieve the fastest running speed. Because the interface with the database also uses this method, it uses localized calls, which also allows the database to exert its best performance.

3. Good safety. Because the code of php itself is open, its code has been tested in the hands of many engineers. At the same time, the way it is compiled with apache also allows it to have flexible security settings. So up to now, php has a recognized security performance.

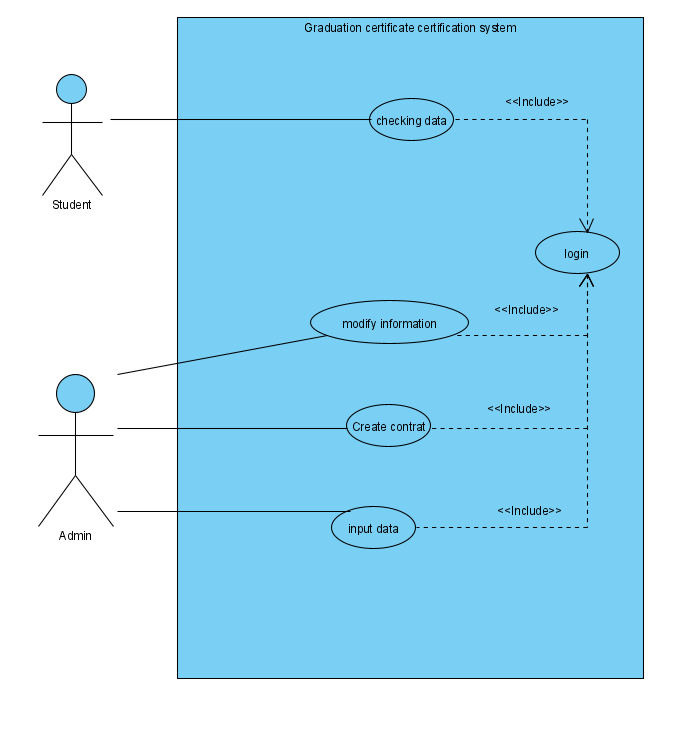
# Design

This chapter will introduce the preliminary design of the whole system. It will be divided into five sections. Section 3.1 will give the overall system architecture. Section 3.2 will list out all the functions required for the system. Section 3.3 will focus on database design, followed by the design of the web application in Section 3.4 respectively. At the end, backend design will be presented in Section 3.5.

## System architecture

This system uses html as the operating interface and uses JavaScript’s function library web3 to call the functions in the contract. In the upload contract, the compiled smart contract is used to transfer to the blockchain. The uploader's id is recorded and then based on the blockchain. The contract program lock can modify the contract account. Only the user who uploaded the contract can only be the user who uploaded the contract. In the upload and update of the data, the user who uploads the account can only be allowed to modify the account. Therefore, the person who modifies the contract can only be the party who uploaded the contract. Modifying the contract party needs to pay a certain amount of gas to execute the operation of the contract. On the reviewing party, the reading function is based on the read-only program due to the contract setting. Once the reader does not need to use gas to access all data in the contract.

## Functional Design



#### Figure 3.1 Use Case Diagram

According to Figure 3.1, this application has a total of five functions, including checking data, login system, modify information, create contract and input data. Among them, we only have two types of users including admin and user.

checking data - to show users whether the data is accurate and prove whether they are graduates of the university.

login system - Identifies the user's information to provide appropriate information and functions.

modify information - modify graduate information with errors.

create contract - Create a storage location.

input data - Enter the graduate information to prove that you are a graduate of this school.

## Database Design

phpMyAdmin will be used for database management. To support content in Traditional Chinese and Simplified Chinese, all collection encoding is set as UTF8.



#### Figure 3.2 Entity Relationship Diagram

The following parameters shown in Figure 3.2 are stored in database and referenced in various application logics.

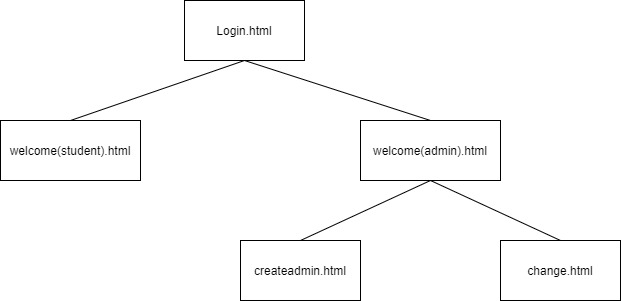
## Blockchain Design

In the blockchain, we will save the student's graduation year, student number, name, and course name. The following parameters shown in table 3.1 are stored in database and referenced in various application logics.

|  |  |  |
| --- | --- | --- |
| # | parameter | Remarks |
| 1 | grad\_year | student graduation year |
| 2 | stud\_id | student number |
| 3 | stud\_name\_zh | Student name Chinese |
| 4 | stud\_name\_en | Student name English |
| 5 | course\_zh | Course name Chinese |
| 6 | course\_en | Course name English |

##### Table 3.1 Blockchain Design Overview

## Web Application Design



#### Figure 3.3 the structure of the entire website

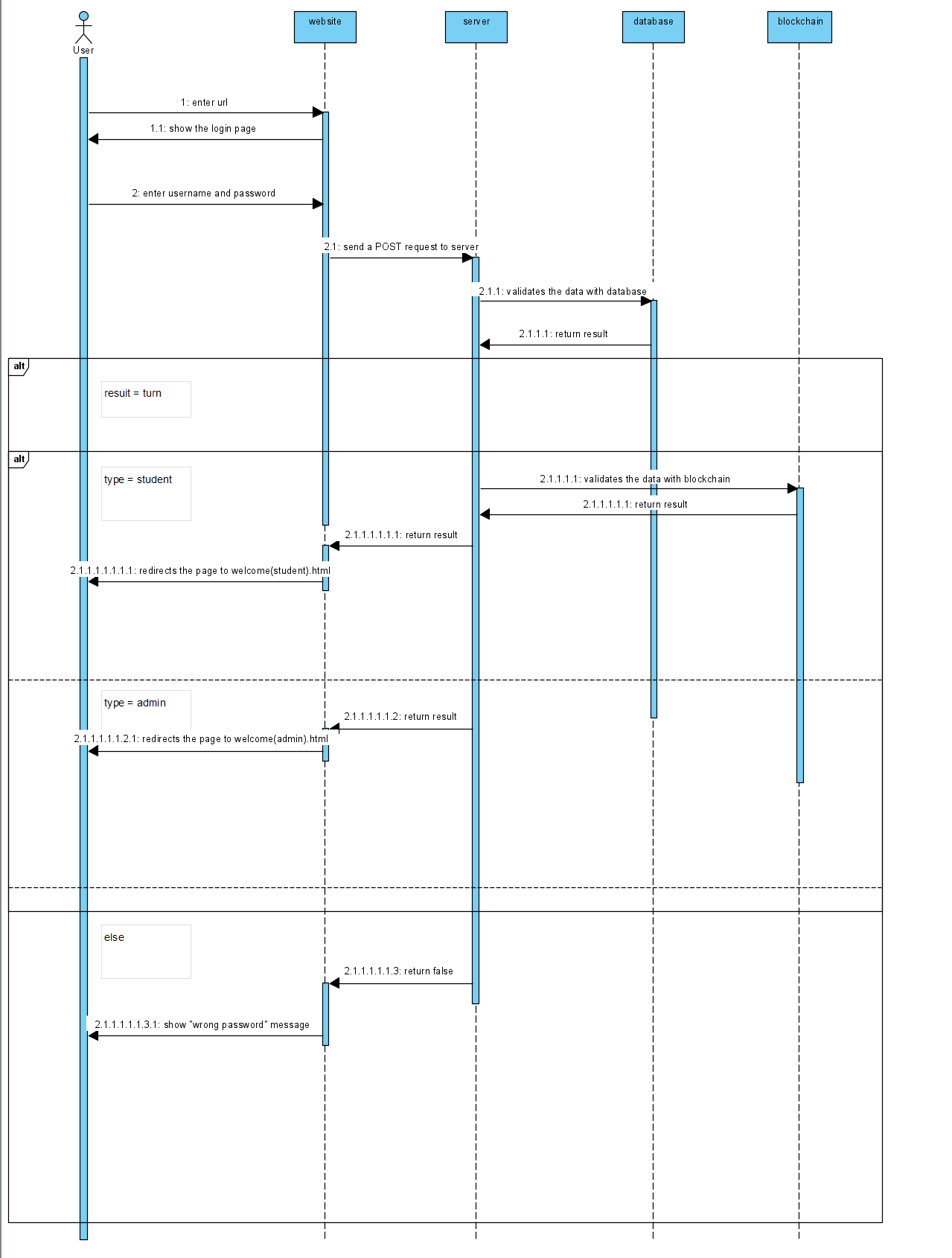
The website application targets current OUHK students. Access control is implemented to safeguard data or system functions from being tampered or misuse. Figure 3.1 shows that the authenticated students and admin on this application. the authenticated students is only can check the graduation information, while admin can modify information, create contract, input graduation student data and create admin.

Figure 3.3 show that the structure of the entire website. it can be further divided into five modules, which are Login, show student data, create student date, create admin, change student data. Each module performs independent and unique functionalities.

### Login

Login is the login page for unauthenticated users. They can log in with their credentials to access other modules or view the public module information center.

The application name will be placed on the top, followed by, input form and buttons. Besides, a validation will be conducted, and a show "wrong password" will be shown when the authentication failed. The logical design authentication is illustrated in Figure 3.4.



#### Figure 3.4 Sequence diagram for login

1. User enter url
2. Website display login page
3. User enters his/her username and password and click login button
4. Website send a POST request to server
5. Server validates the data with database
6. Database return result
7. If result is turn and the type is student, Server validates the data with blockchain and website display welcome(student).html
8. If result is turn and the type is admin, website display welcome(admin).html
9. If result is false, website display “wrong password” message.

### create student date

Through this function, you can upload new students' graduation information to the blockchain for permanent stay and give students a certificate of graduation.

### show student data

this function can help user to check the information about graduation. it only shows for the student. Also, it shows the student graduation information when the student login successful.

### create admin

After the user selects the create admin button on the main page, the system will display the create admin page. This page can be used to create a new admin account for admin to use the site.

### change student data

Through this function, you can upload changed students' graduation information to the blockchain for permanent stay and give students a certificate of graduation.

## Backend Design

The backend is written in PHP. The usage of the files is presented in Table 3.2.

|  |  |  |  |
| --- | --- | --- | --- |
| # | File name | Module | Remarks |
| 1 | connect.php | General | Database Configuration |
| 2 | create\_user.php | Create admin  Create student date | - |
| 3 | login.php | login | - |

##### Table 3.2 Backend Files Overview

The following Table 3.3 shows the functions and the required parameters in each PHP file.

|  |  |  |
| --- | --- | --- |
| #1 connect.php | | |
| # | Functions | Parameters required |
| 1 | connect | - |
| #2 create\_user.php | | |
| # | Functions | Parameters required |
| 1 | create\_user | stud\_id, pw, blockchain\_code, type |
| #3 login.php | | |
| # | Functions | Parameters required |
| 1 | login | username, pw |

##### Table 3.3 Backend Functions Overview

# Experiment

In this section, the development details of blockchain will be described.

## Create blockchain server

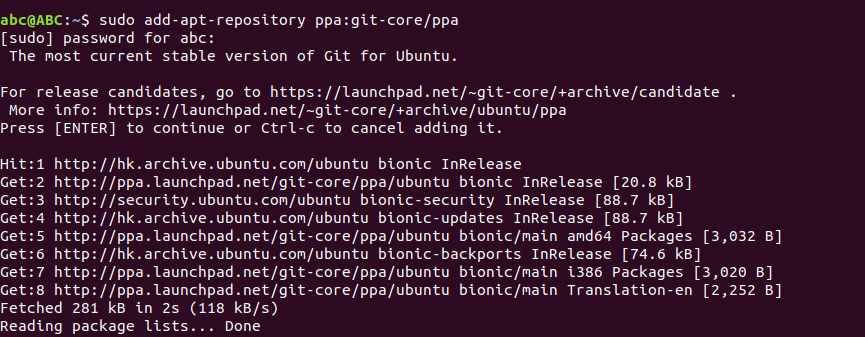
The system is built on the Ubuntu 18.04 LTS operating system. The blockchain system uses geth and the private Ethereum blockchain for experiments. The decentralized application (DApp) development tools used in this study use JavaScript, including the web3 library. The following subsections show implementation and evaluation.

### install git

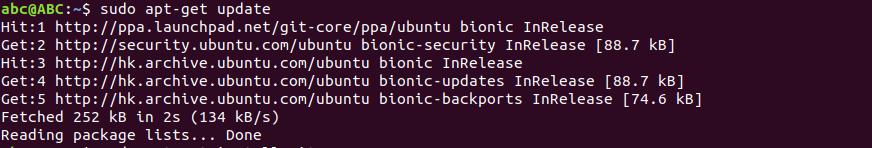
We can install git through table 4.1 command. Git is a distributed version control software designed to better manage Linux core development. We can use git independently for version control. Figure 4.1 to 4.3 respectively indicate the results obtained by the command.

|  |  |  |
| --- | --- | --- |
| # | command | Remarks |
| 1 | abc@ABC:~$ sudo add-apt-repository ppa:git-core/ppa | Install git |
| 2 | abc@ABC:~$ sudo apt-get update |
| 3 | abc@ABC:~$ sudo apt-get install git |

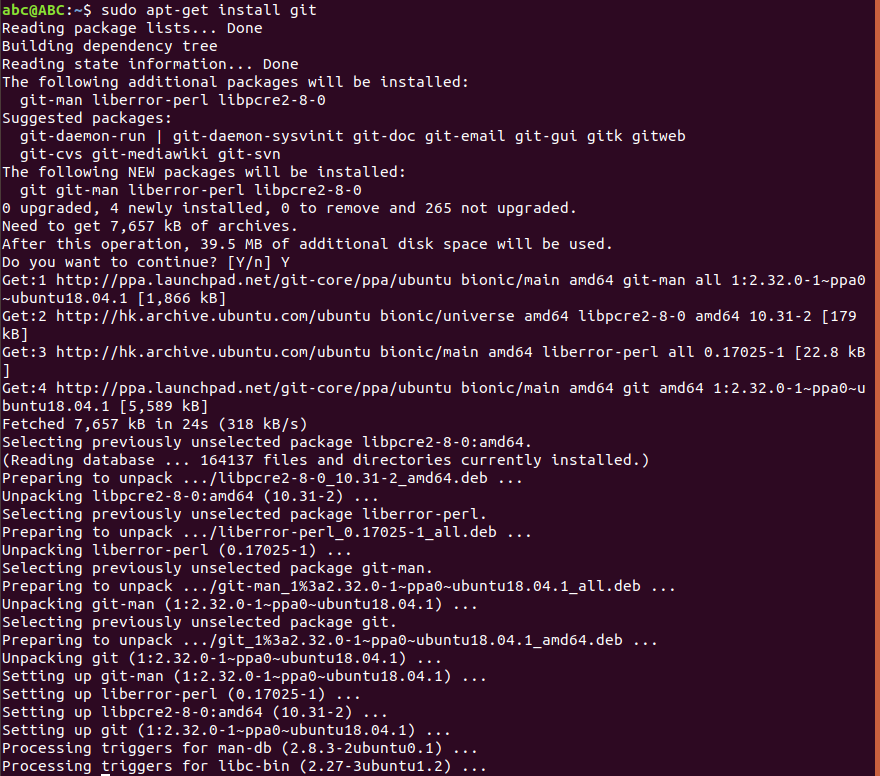
##### Table 4.1 Install git command



#### Figure 4.1 command#1 result



#### Figure 4.2 command#2 result



#### Figure 4.3 command#3 result

### Check git version

We can use the command of table 4.2 to check whether we have installed git. Figure 4.4 respectively indicate the results obtained by the command.

|  |  |  |
| --- | --- | --- |
| # | command | Remarks |
| 4 | abc@ABC:~$ git --version | Check git version |

##### Table 4.2 Check git version command



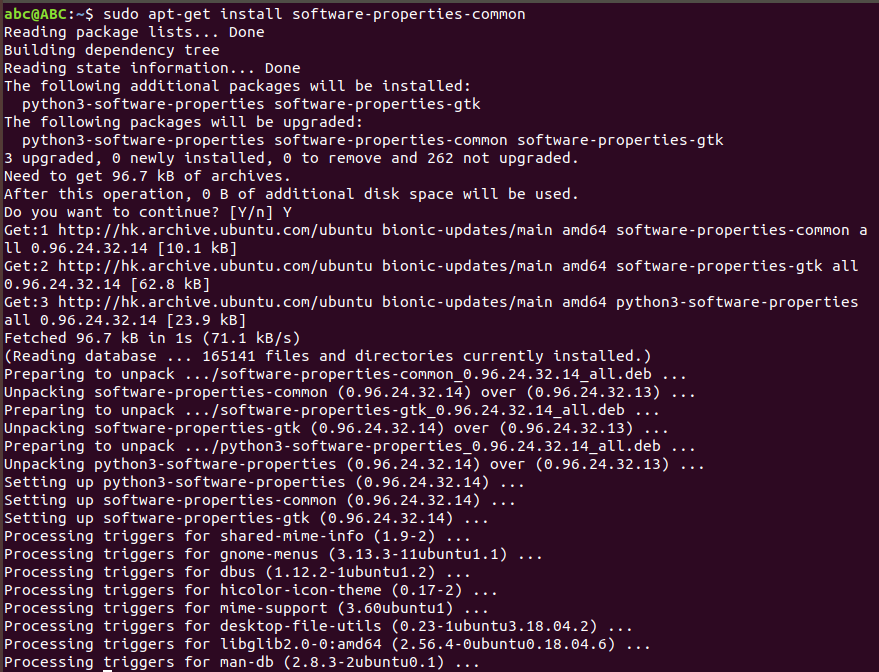
#### Figure 4.4 command#4 result

### Install geth

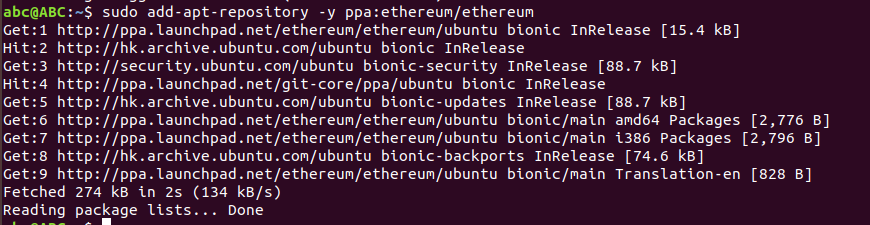
We can install git through table 4.3 command. Geth is Go Ethereum’s standalone CLI client and is the most popular software client for running a node on the Ethereum network. Running a node allows users to perform transactions and interact with smart contracts on the Ethereum blockchain. Figure 4.5 to 4.8 respectively indicate the results obtained by the command.

|  |  |  |
| --- | --- | --- |
| # | command | Remarks |
| 5 | abc@ABC:~$ sudo apt-get install software-properties-common | Install geth |
| 6 | abc@ABC:~$ sudo add-apt-repository -y ppa:ethereum/ethereum |
| 7 | abc@ABC:~$ sudo apt-get update |
| 8 | abc@ABC:~$ sudo apt-get install ethereum |

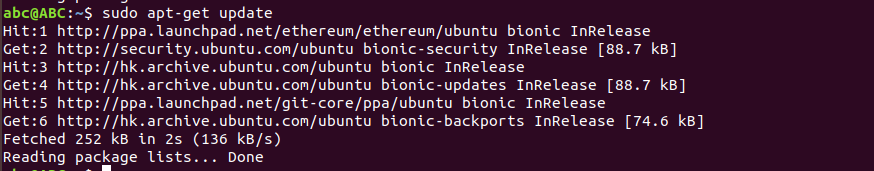
##### Table 4.3 Install geth command



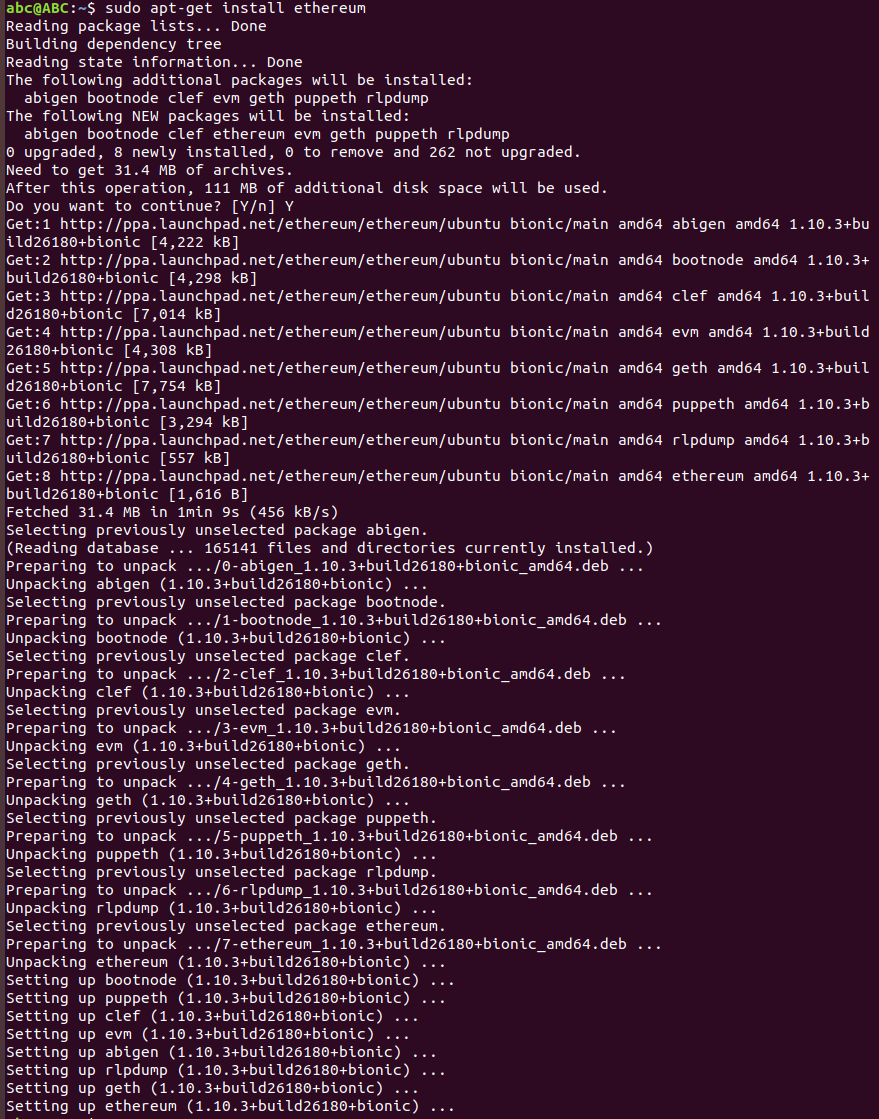
#### Figure 4.5 command#5 result



#### Figure 4.6 command#6 result



#### Figure 4.7 command#7 result



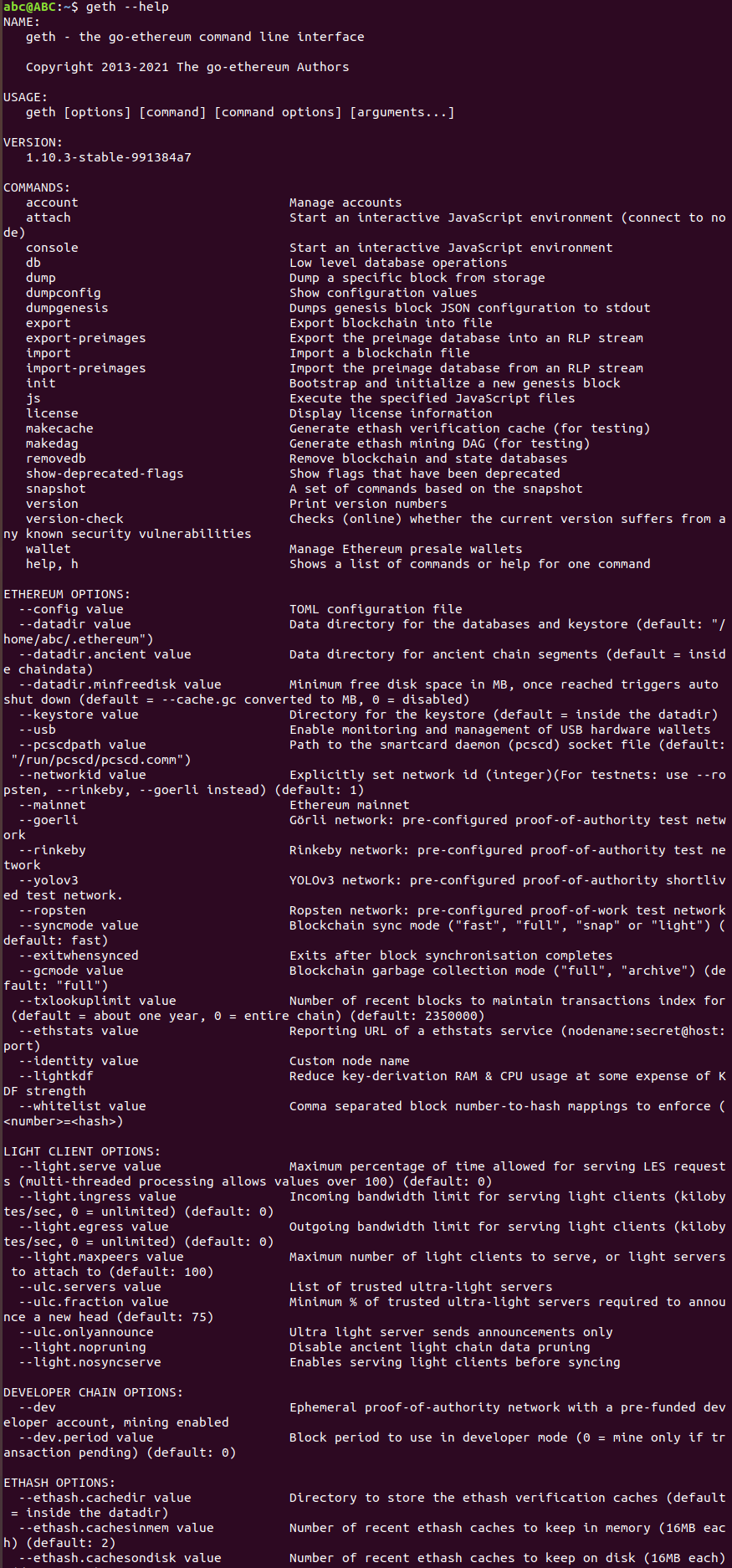
#### Figure 4.8 command#8 result

### check geth

We can use the command of table 4.4 to check whether we have installed git. Figure 4.7 respectively indicate the results obtained by the command.

|  |  |  |
| --- | --- | --- |
| # | Command | Remarks |
| 9 | abc@ABC:~$ geth –help | Check geth |

##### Table 4.4 check geth command



#### Figure 4.9 command#9 result

## build a private blockchain

### Create a folder to store private blockchain data files

In table 4.5, it show that create a new file command.

|  |  |  |
| --- | --- | --- |
| # | command | Remarks |
| 10 | abc@ABC:~$ cd /home/abc | Open folder |
| 11 | abc@ABC:~$ mkdir private\_chain | Create a new folder |
| 12 | abc@ABC:~$ cd private\_chain | Open folder |

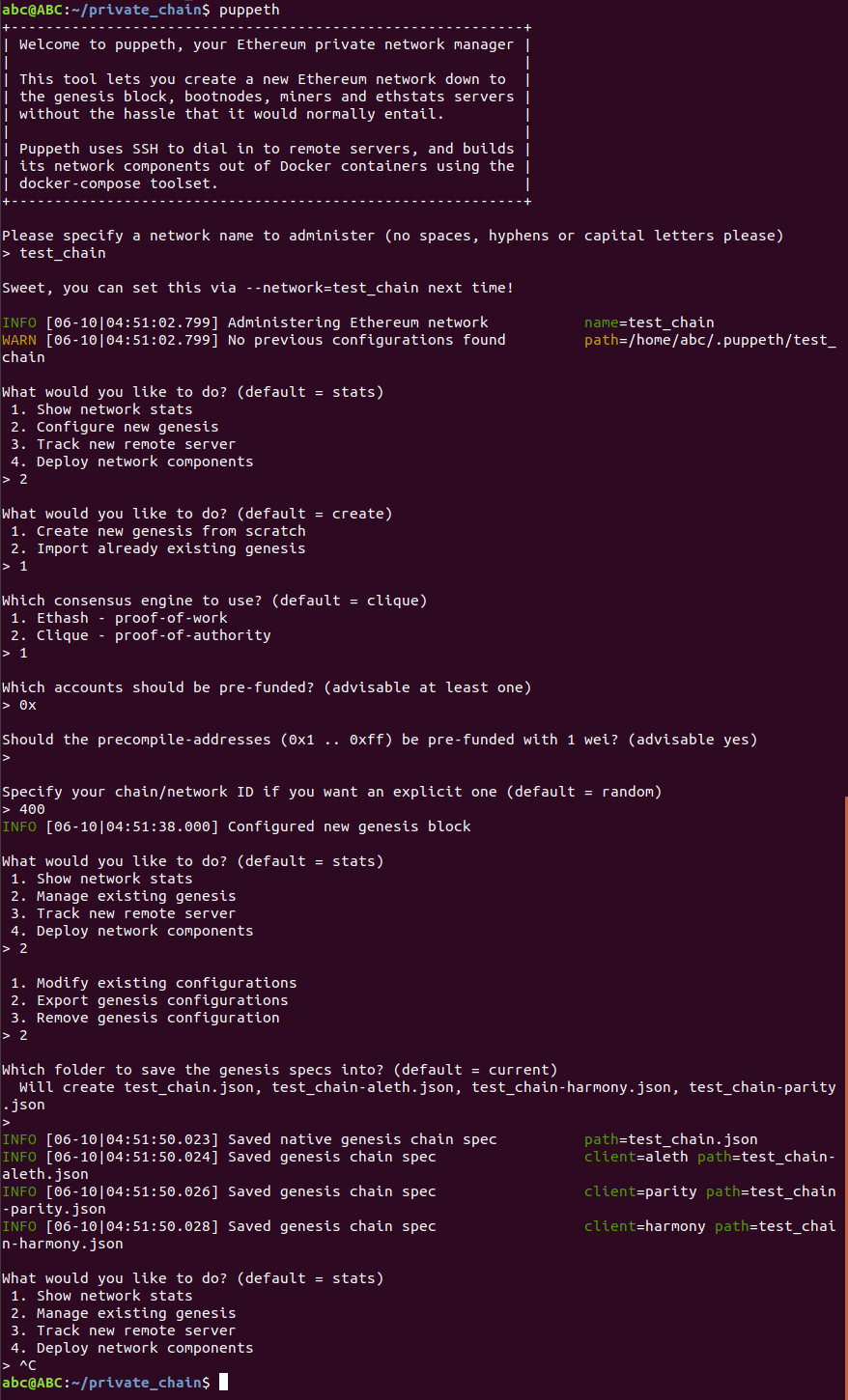
##### Table 4.5 Create file command

### generate genesis.json file

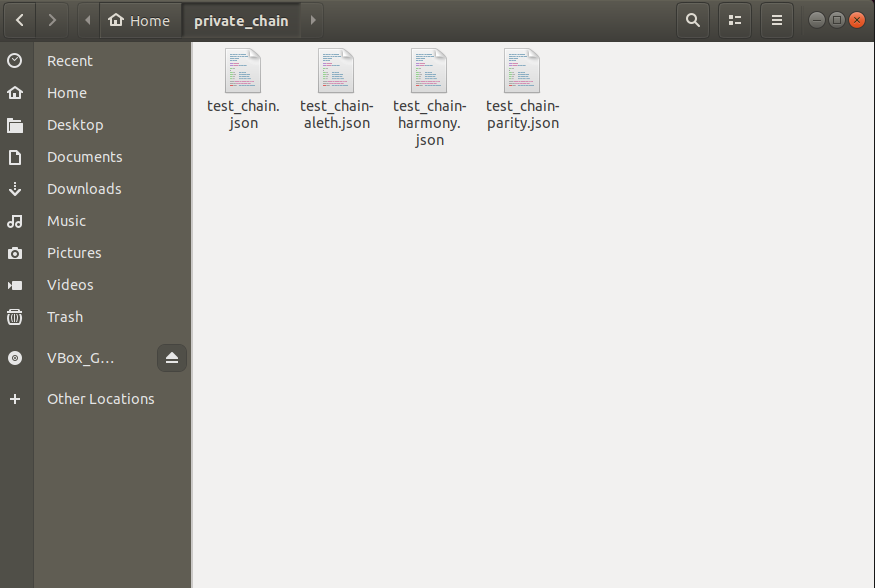
In table 4.6, it show that generate genfsis gensis.json file. Figure 4.10 show that the command#13 result. Figure 4.11 show that generate 4 json file for gensis block.

|  |  |  |
| --- | --- | --- |
| # | command | Remarks |
| 13 | abc@ABC:~/private\_chain$ puppeth  +-----------------------------------------------------------+  | Welcome to puppeth, your Ethereum private network manager |  | |  | This tool lets you create a new Ethereum network down to |  | the genesis block, bootnodes, miners and ethstats servers |  | without the hassle that it would normally entail. |  | |  | Puppeth uses SSH to dial in to remote servers, and builds |  | its network components out of Docker containers using the |  | docker-compose toolset. |  +-----------------------------------------------------------+  Please specify a network name to administer (no spaces, hyphens or capital letters please)  > test\_chain  Sweet, you can set this via --network=test\_chain next time!  INFO [06-10|04:51:02.799] Administering Ethereum network name=test\_chain  WARN [06-10|04:51:02.799] No previous configurations found path=/home/abc/.puppeth/test\_chain  What would you like to do? (default = stats)  1. Show network stats  2. Configure new genesis  3. Track new remote server  4. Deploy network components  > 2  What would you like to do? (default = create)  1. Create new genesis from scratch  2. Import already existing genesis  > 1  Which consensus engine to use? (default = clique)  1. Ethash - proof-of-work  2. Clique - proof-of-authority  > 1  Which accounts should be pre-funded? (advisable at least one)  > 0x  Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)  >  Specify your chain/network ID if you want an explicit one (default = random)  > 400  INFO [06-10|04:51:38.000] Configured new genesis block  What would you like to do? (default = stats)  1. Show network stats  2. Manage existing genesis  3. Track new remote server  4. Deploy network components  > 2  1. Modify existing configurations  2. Export genesis configurations  3. Remove genesis configuration  > 2  Which folder to save the genesis specs into? (default = current)  Will create test\_chain.json, test\_chain-aleth.json, test\_chain-harmony.json, test\_chain-parity.json  >  INFO [06-10|04:51:50.023] Saved native genesis chain spec path=test\_chain.json  INFO [06-10|04:51:50.024] Saved genesis chain spec client=aleth path=test\_chain-aleth.json  INFO [06-10|04:51:50.026] Saved genesis chain spec client=parity path=test\_chain-parity.json  INFO [06-10|04:51:50.028] Saved genesis chain spec client=harmony path=test\_chain-harmony.json  What would you like to do? (default = stats)  1. Show network stats  2. Manage existing genesis  3. Track new remote server  4. Deploy network components  > ^C | generate genesis.json file |

##### Table 4.6 generate genesis.json file command



#### Figure 4.10 command#13 result



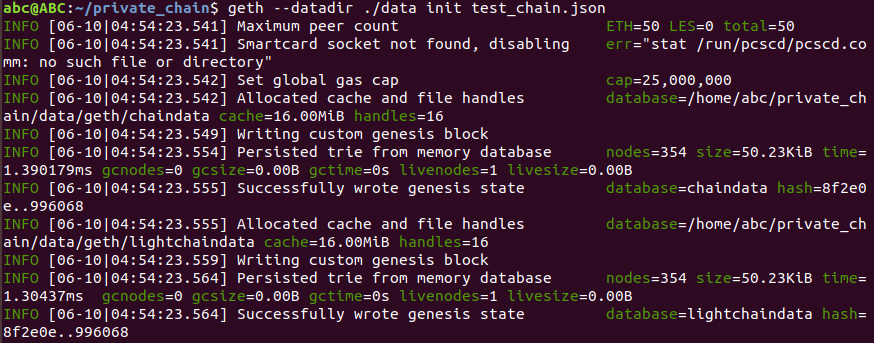
#### Figure 4.11 private\_chain folder

### create a genesis block

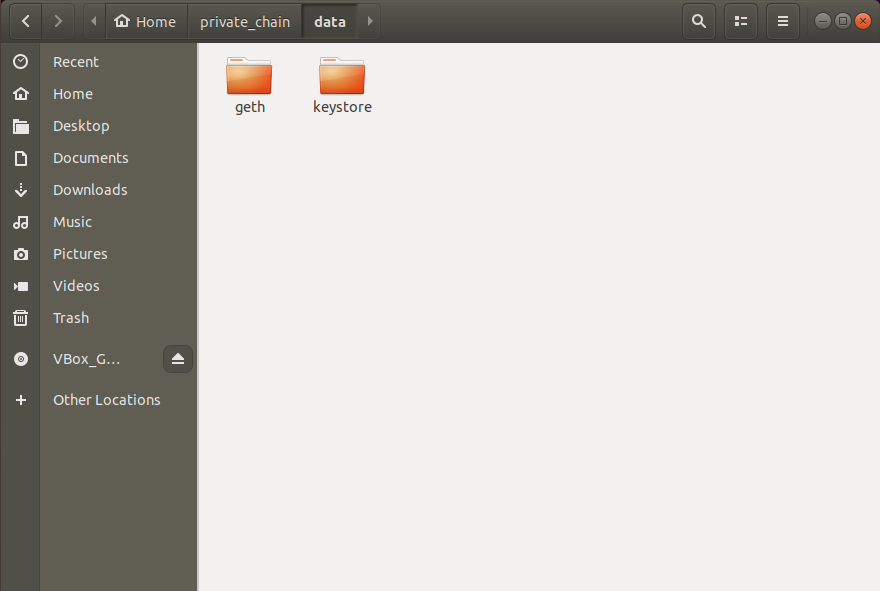
in table 4.7, it show that create a gensis block command. In command 14, it will create a new folder “data” to store the data generated by the private chain. In command 15 and figure 4.12, it will generate the genesis block. In command 16 and figure 4.14, it will go enter the blockchain console.

|  |  |  |
| --- | --- | --- |
| # | command | Remarks |
| 14 | abc@ABC:~/private\_chain$ mkdir data | store the data generated by the private chain |
| 15 | abc@ABC:~/private\_chain$ geth --datadir ./data init test\_chain.json | generate the genesis block |
| 16 | abc@ABC:~/private\_chain$ geth --port 30303 --rpc --rpcaddr 0.0.0.0 --rpccorsdomain="\*" --rpcport 8100 --ws --allow-insecure-unlock --syncmode "full" --nodiscover --maxpeers 1000 --maxpendpeers 1000 --networkid 400 --datadir ./data console | enter the console |

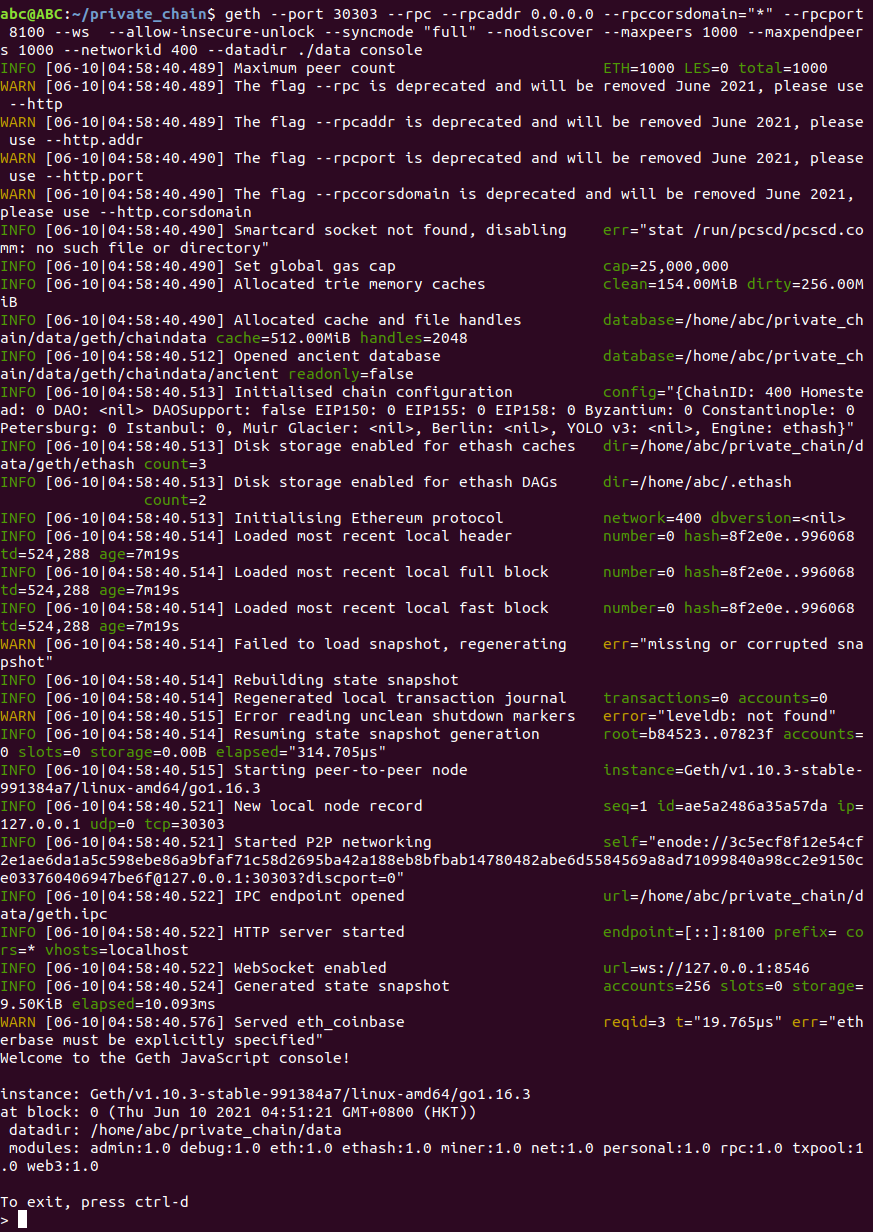
##### Table 4.7 create a genesis block command



#### Figure 4.12 command#15 result



#### Figure 4.13 data folder



#### Figure 4.14 command#16 result

## testing blockchain

In table 4.8, it show that testing blockchain command. Figure 4.15 and 4.16 show that returns the current block detail. Figure 4.17 show that to create new a account in blockchain. Figure 4.18 show that it mining successful.

|  |  |  |
| --- | --- | --- |
| # | command | Remarks |
| 17 | > eth.getBlock(0) | - |
| 18 | > eth.blockNumber | - |
| 19 | > eth.mining | - |
| 20 | > net.version | - |
| 21 | > net.peerCount | - |
| 22 | > personal.newAccount() | - |
| 23 | > miner.start(1) | - |

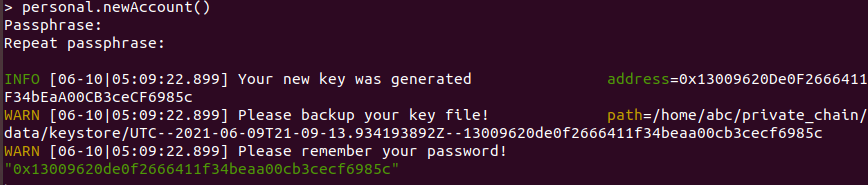
##### Table 4.8 testing blockchain command



#### Figure 4.15 command#17 result



#### Figure 4.16 command#18-21 result



#### Figure 4.17 command#22 result



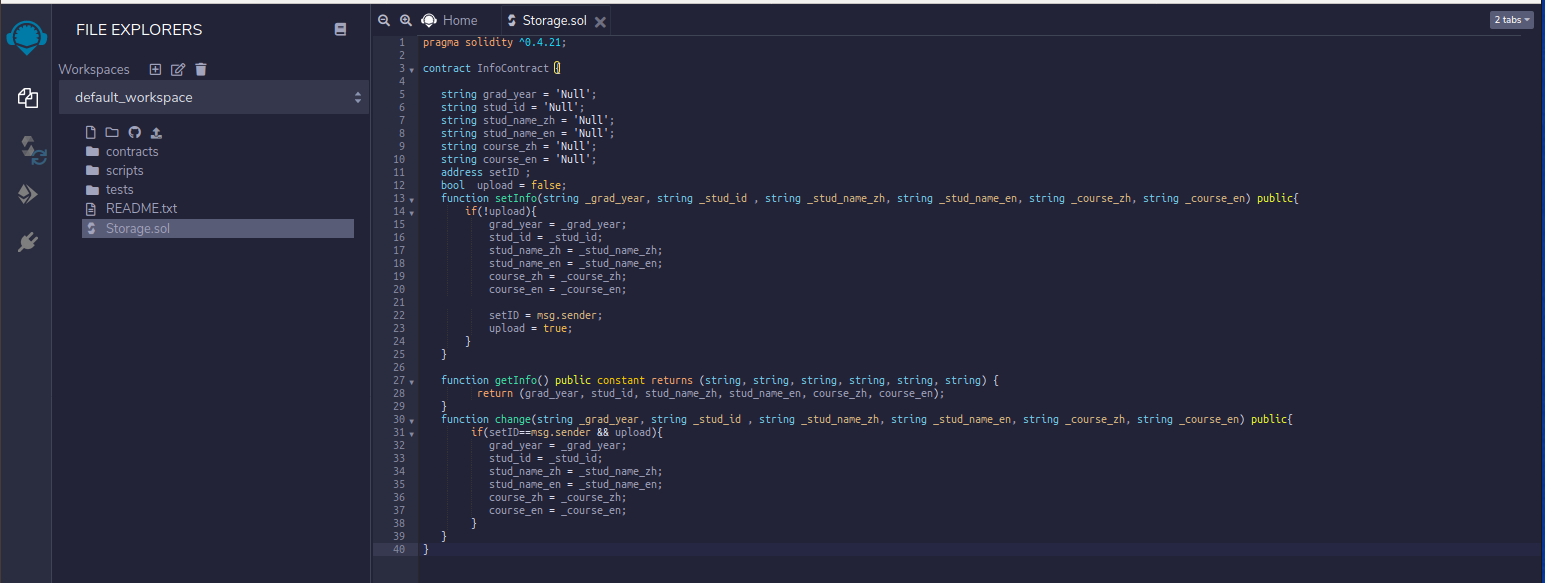
#### Figure 4.18 command#23 result

## Smart contract

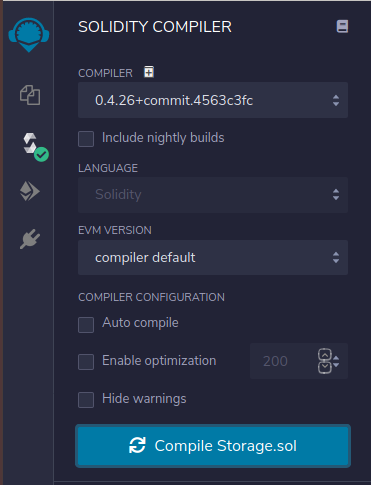
In smart contract, I use the Remix IDE to create and connect the blockchain.

Remix IDE is an open-source web and desktop application. It fosters a fast development cycle and has a rich set of plugins with intuitive GUIs. Remix is used for the entire journey of contract development as well as being a playground for learning and teaching Ethereum.

In figure 4.19, it show that the remix UI and created the storage.sol for the smart contract.in figure 4.20, it show that go to solidity compiler to compile the file. In figure 4.21, it show that go to deploy and run transactions and connect to the blockchain to upload file. When upload successful, remix will return figure 4.22 result.



#### Figure 4.19 blockchain coding



#### Figure 4.20 building coding

一張含有 文字, 監視器, 黑色, 螢幕擷取畫面 的圖片

自動產生的描述

#### Figure 4.21 upload coding to blockchain

一張含有 文字 的圖片

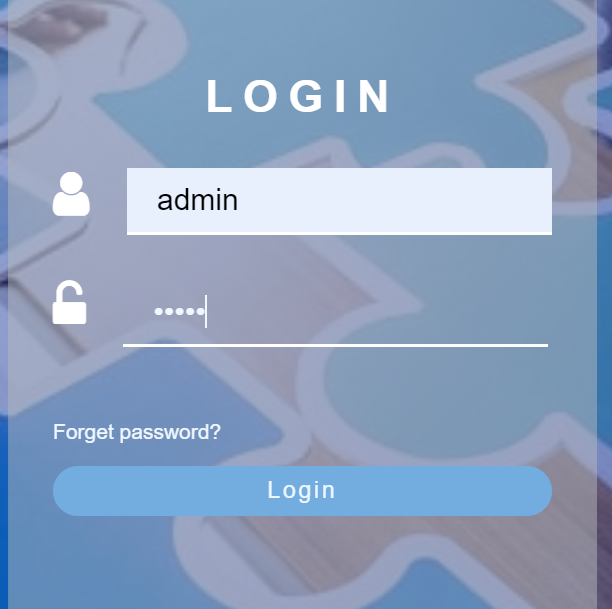
自動產生的描述

#### Figure 4.22 upload coding successful to blockchain

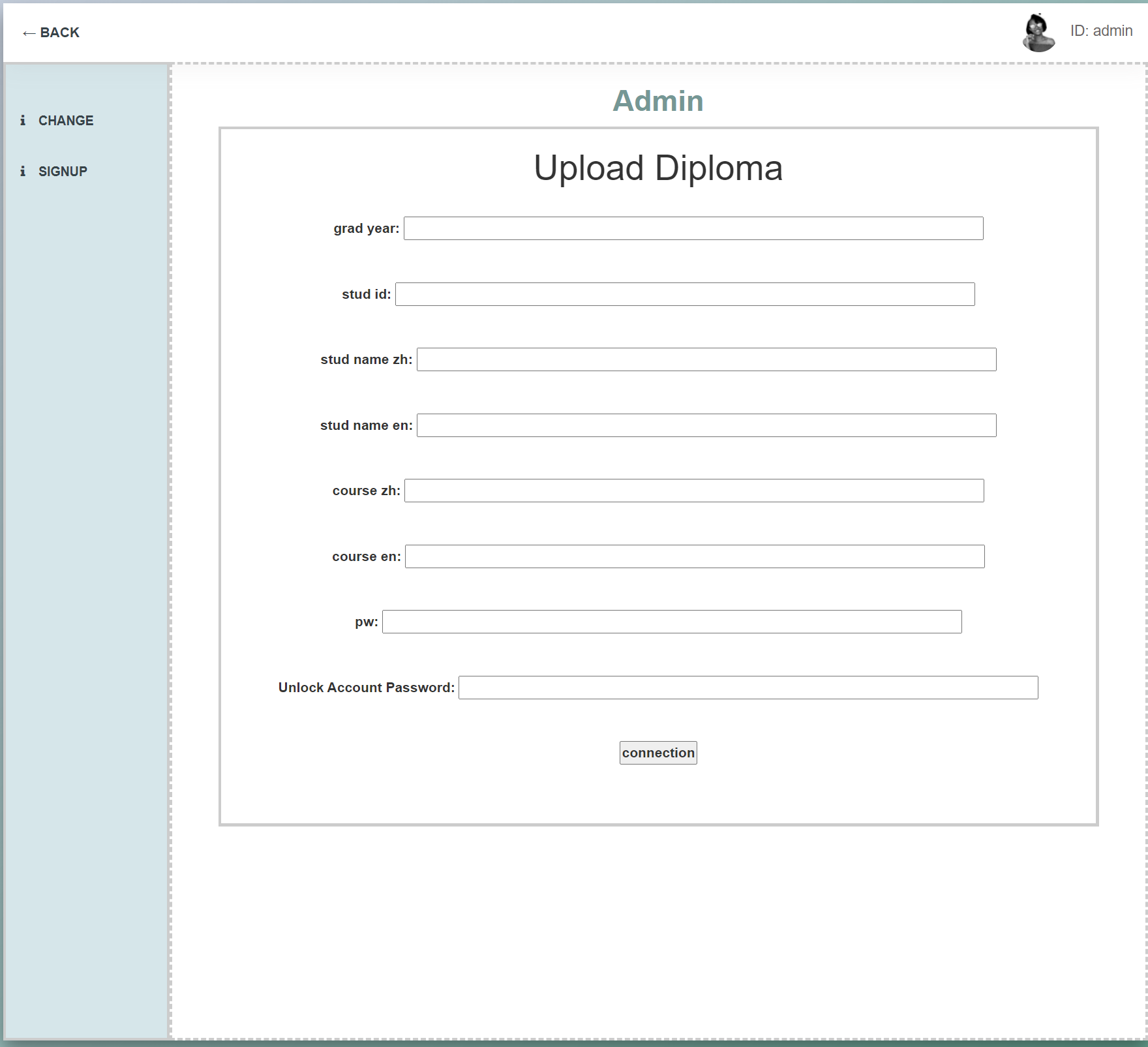
# Results and Discussion

## Login

For the login page, as shown in Figure 5.1, the user is allowed to enter the username and password, and then click the user and administrator button to log in to different roles. The user and administrator will have different functions after logging in. Figure 5.2 show that the admin login successful.



#### Figure 5.1 login page



#### Figure 5.2 welcome(admin).html

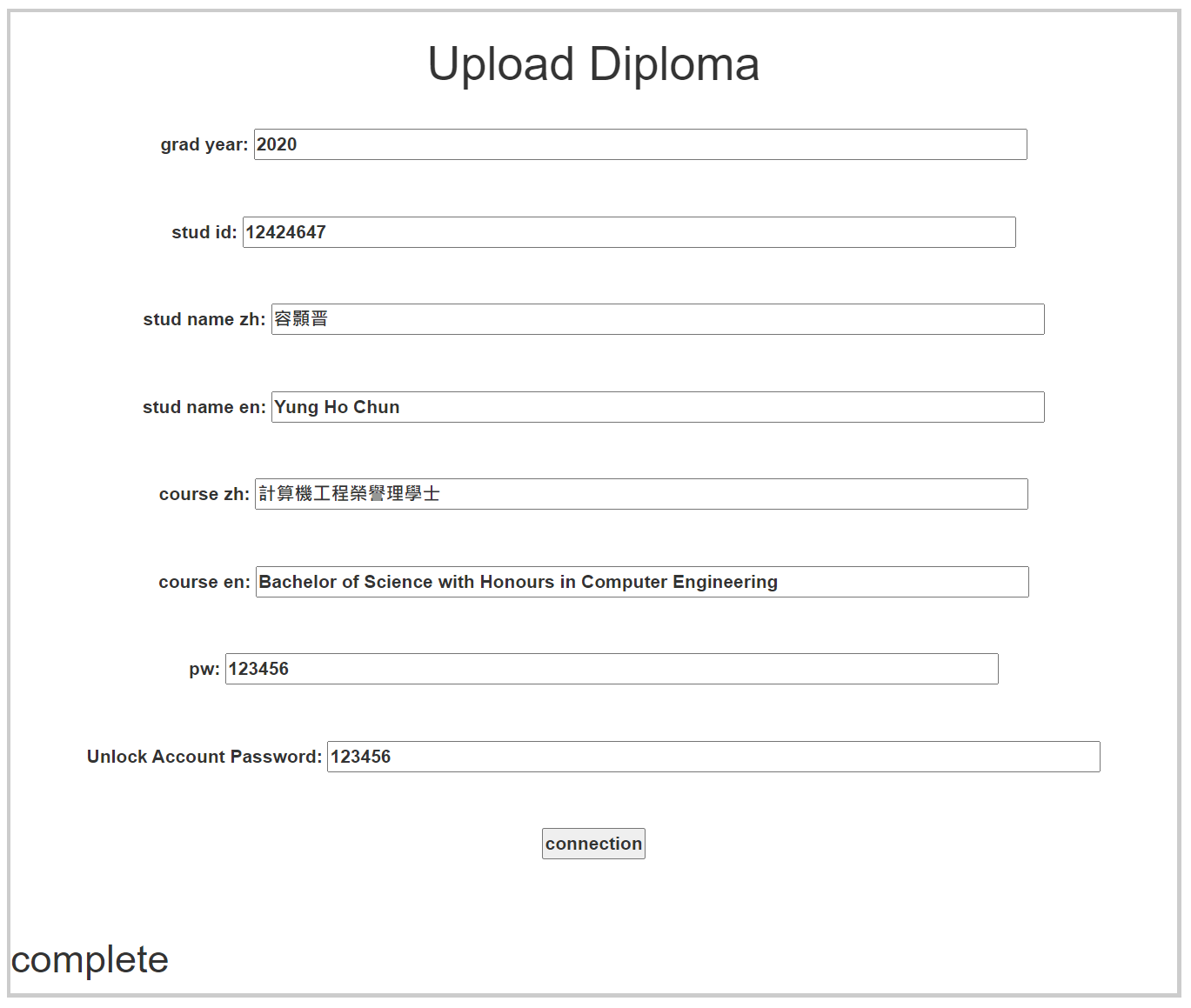
## create and show student date

Figure 5.3 show that the upload student graduation information successful in website. Also,

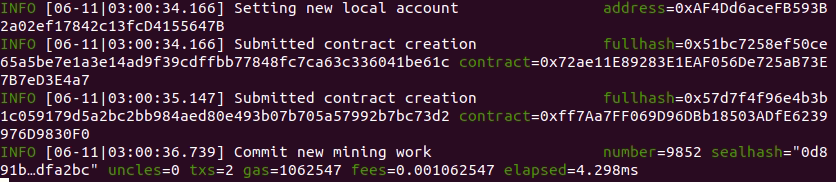
In figure 5.4, The first line indicates that the account has been successfully logged in and setting a local account for the new student graduation information. The second item is the address of the smart contract on the blockchain. The contract is particularly important. We can search for the entire full hash value by recording the contract. figure 5.6 show that the database is recorded the contract to give the student use. The third line is the proof of the workload of the entire record, and it is stored in the contract The last line is to continue the mining process.

figure 5.5 and 5.6 show that the upload student graduation information successful in blockchain.

Through this function, we have applied the blockchain. It is because we can use this function to store student graduation information and upload it to the blockchain through smart contracts, so that student information can be protected.



#### Figure 5.3 enter student information



#### Figure 5.4 upload data in blockchain



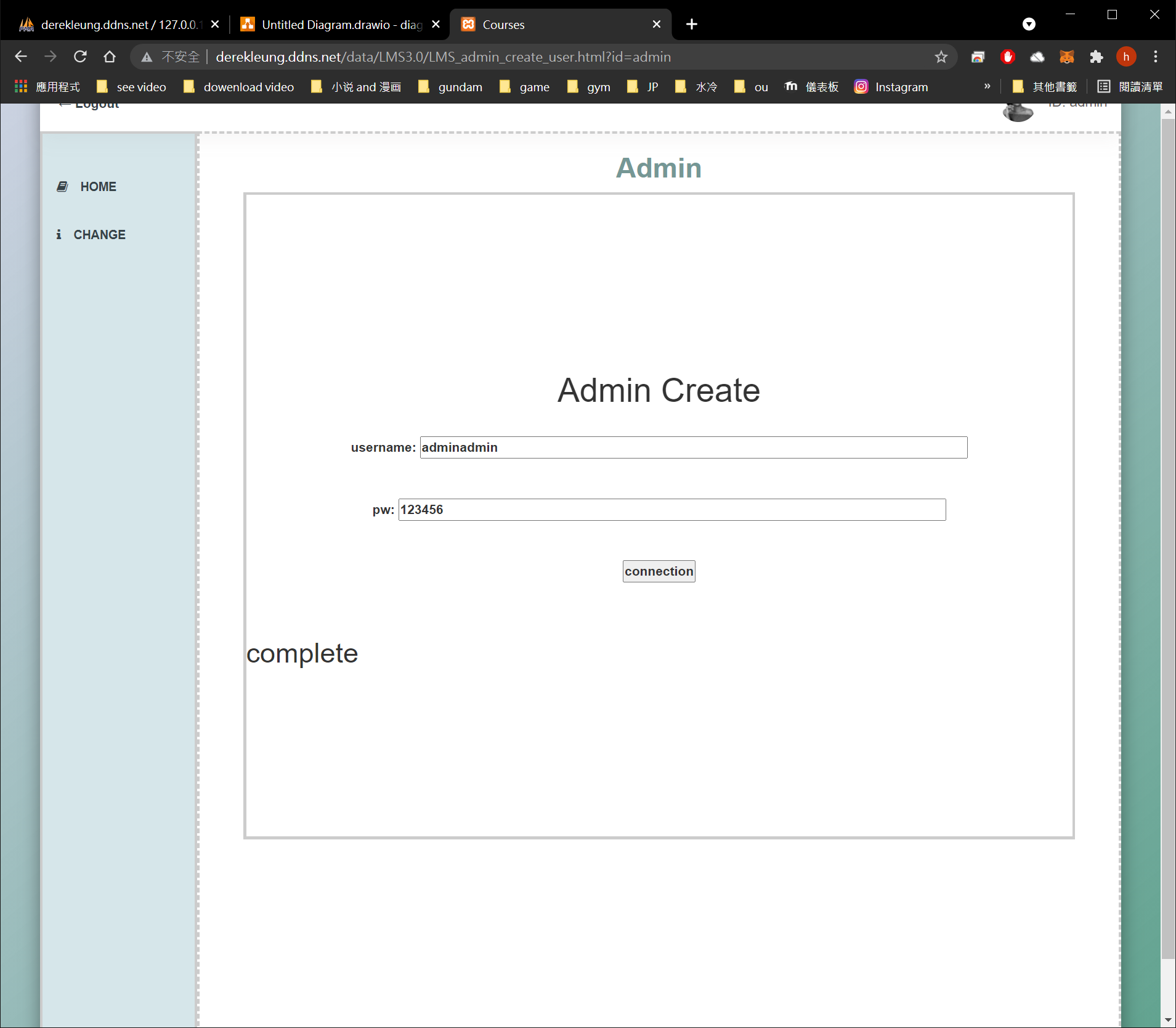
#### Figure 5.5 database date



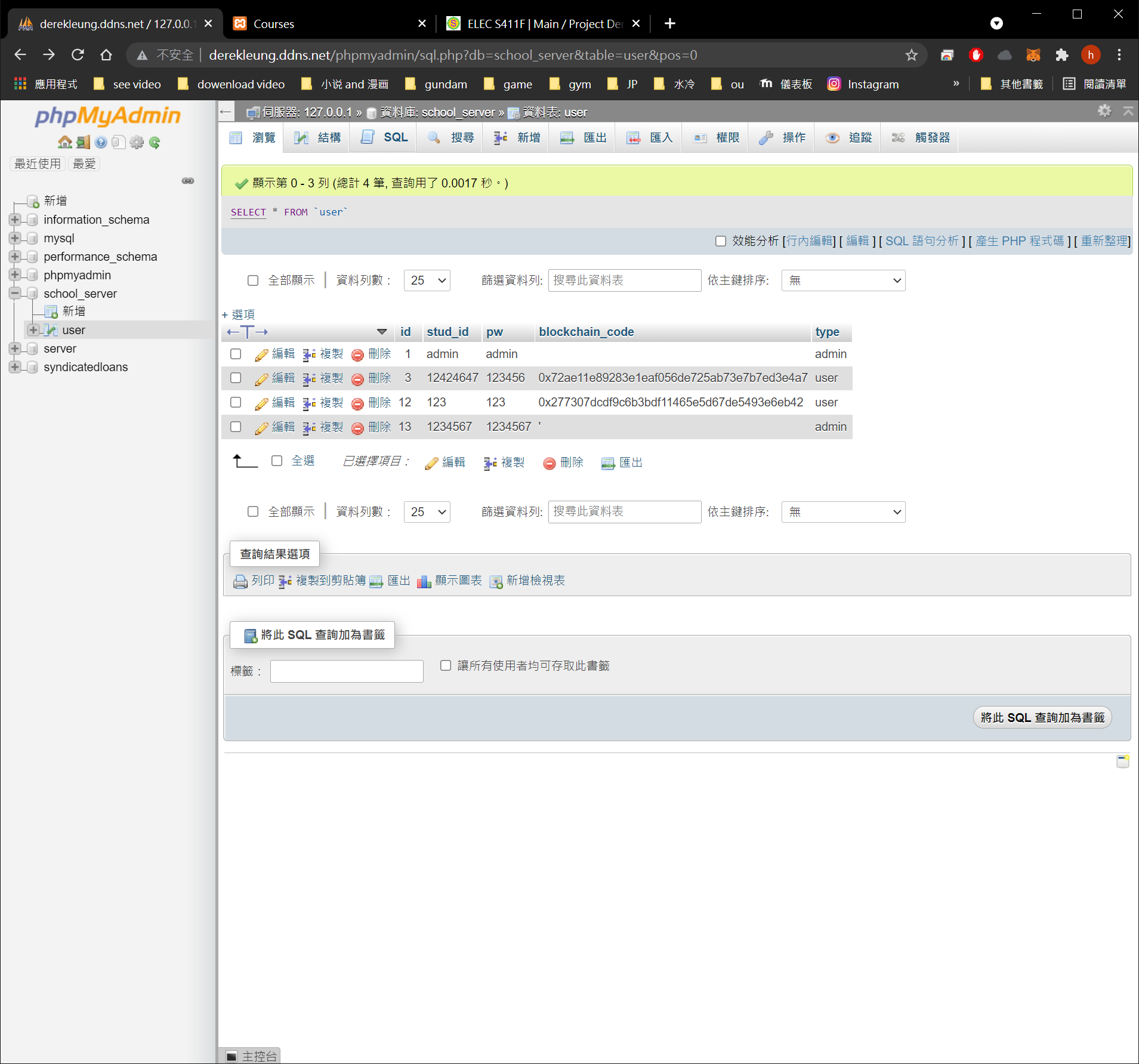
#### Figure 5.6 display student graduation information

## create admin

If created successfully (Figure 5.7), the system will automatically generate an ID for the new account. This is because of account security issues, because users can not register their own account, but the ID as login information can’t be the user's own personal information, because colleagues or other people in the company may have the opportunity to know the data through the user's personal information. Therefore, choose to automatically generate a set of numbers as the ID (Figure 5.8).



#### Figure 5.7 create admin data

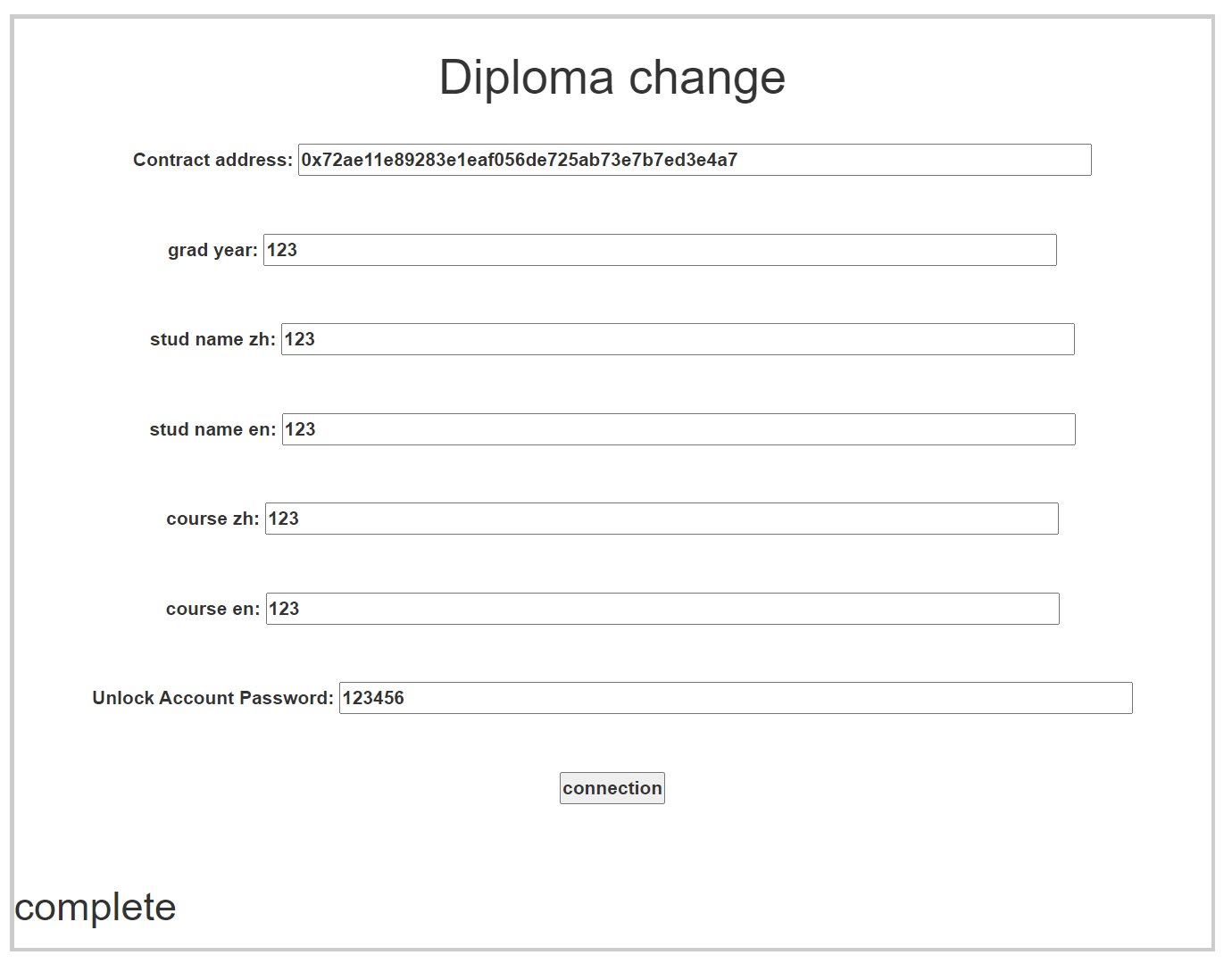


#### Figure 5.8 create account successful

## change student date

Through this function, we have applied the blockchain. Because we can use this function to find the corresponding hash value on the blockchain, modify the student's graduation information (figure 5.9), and upload it to the blockchain through a smart contract, so that the student's information can be protected.

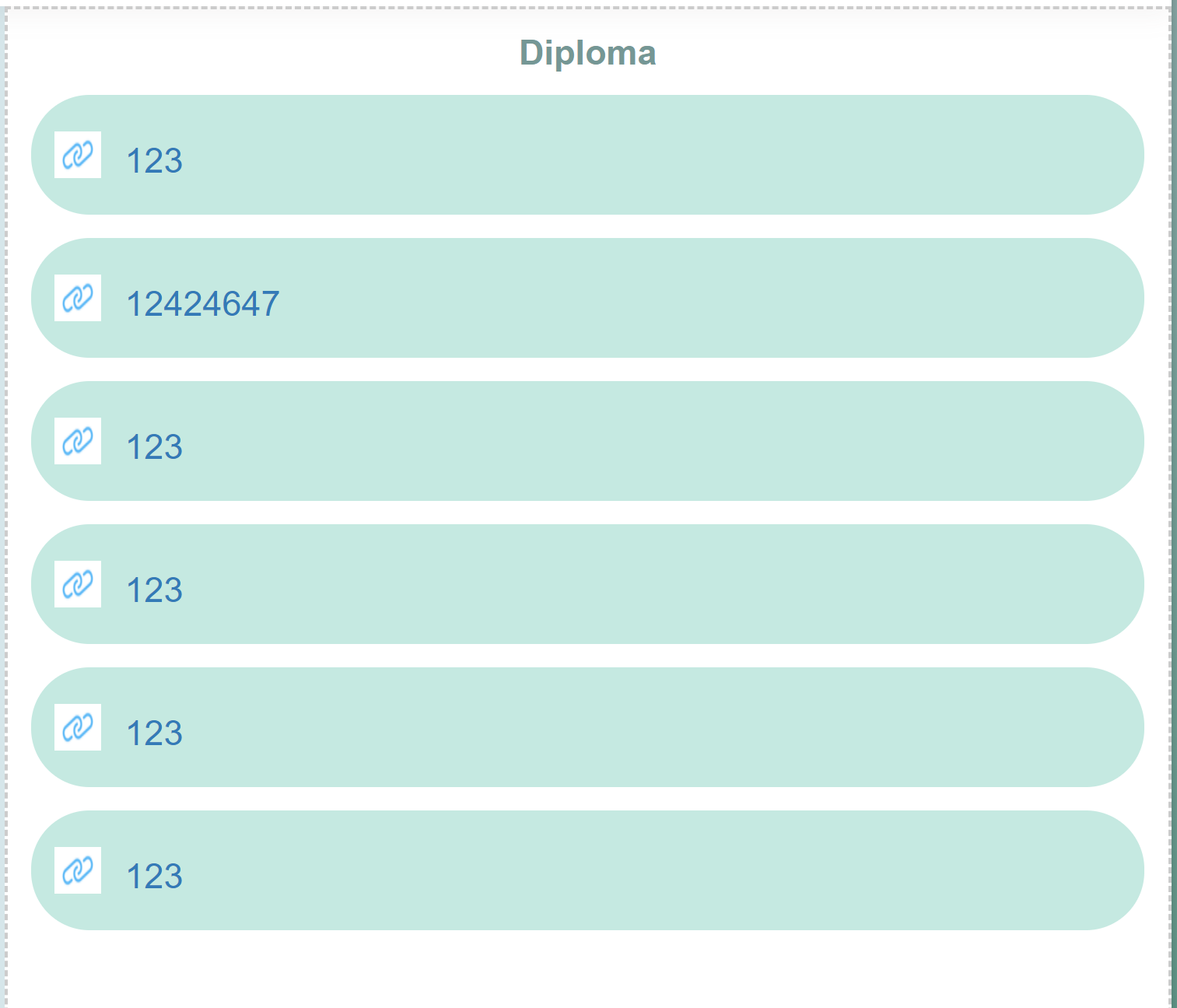
In figure 5.10, it show that we successfully changed the corresponding hash value on the blockchain, and the blockchain also recorded its proof of work. In figure 5.11, it show that display successful changed data.



#### Figure 5.9 change student information successful



#### Figure 5.10 upload data in blockchain



#### Figure 5.11 display student graduation information

# Conclusion and Further Work

To summarize this report, this project uses the blockchain to store the graduation information of university students. The functions of the system include storing the graduation information of university students, displaying the graduation information of university students, modifying the graduation information of university students, and adding administrators. The above system functions have been developed. Although the user-friendly operation has not been achieved, the core functions have been completed.

The users of this system are designed for students and administrators to make it easier for administrators to enter a large number of graduate students' information, and for students to more easily prove that their graduation certificates are not fake. The core goal has been completed, and students can prove their academic qualifications to employers anytime and anywhere. At the same time, the administrator can complete the work faster. In addition, because the administrator has the opportunity to input the wrong graduate student's information when entering a large amount of input, the system also has a system for revising the graduate student's information. This goal has also been accomplished.

Of course, there are also imperfections in the project, which can still be modified and improved in future work. Although it uses the decentralization of the blockchain and cannot be modified at will, it does not fully utilize the encryption of the blockchain, so the technology of the blockchain can be used again. Although the encryption feature is not necessary in this system, the encryption feature can add more security to the system and make users feel more at ease. So in the future, we will improve this function so that more users can use this system.

# References

1. 鄺曉斌. (2018, August 26). 網上湧現假大學證書 仿真度高 人力顧問、律師亦難辨當假為真. Retrieved March 08, 2021, from <https://www.hk01.com/%E7%A4%BE%E6%9C%83%E6%96%B0%E8%81%9E/206312/%E7%B6%B2%E4%B8%8A%E6%B9%A7%E7%8F%BE%E5%81%87%E5%A4%A7%E5%AD%B8%E8%AD%89%E6%9B%B8-%E4%BB%BF%E7%9C%9F%E5%BA%A6%E9%AB%98-%E4%BA%BA%E5%8A%9B%E9%A1%A7%E5%95%8F-%E5%BE%8B%E5%B8%AB%E4%BA%A6%E9%9B%A3%E8%BE%A8%E7%95%B6%E5%81%87%E7%82%BA%E7%9C%9F>
2. Blockchains: The great chain of being sure about things. The Economist. 31 October 2015 [18 June 2016]. The technology behind bitcoin lets people who do not know or trust each other build a dependable ledger. This has implications far beyond the crypto currency.
3. Morris, David Z. Leaderless, Blockchain-Based Venture Capital Fund Raises $100 Million, And Counting. Fortune. 15 May 2016 [2016-05-23].
4. Popper, Nathan. A Venture Fund With Plenty of Virtual Capital, but No Capitalist. The New York Times. 21 May 2016 [2016-05-23].
5. Brito, Jerry; Castillo, Andrea. Bitcoin: A Primer for Policymakers (PDF). Fairfax, VA: Mercatus Center, George Mason University. 2013 [22 October 2013].
6. Trottier, Leo. original-bitcoin (self-published code collection). github. 18 June 2016 [2016-06-18].
7. Gray, Jeff. Bitcoin believers: Why digital currency backers are keeping the faith. The Globe and Mail (Phillip Crawley). 2014-04-07 [2016-02-17].
8. Vigna, Paul. BitBeat: Microsoft to Offer Ethereum-Based Services on Azure. The Wall Street Journal (Blog). News Corp. 2015-10-28 [2016-02-17].
9. Schueffel, Patrick; Groeneweg, Nikolaj; Baldegger, Rico. The Crypto Encyclopedia: Coins, Tokens and Digital Assets from A to Z. Bern, 瑞士: Growth Publisher. 2019 [2020-02-18].
10. Smart Contracts, Explained. Cointelegraph. 2017-10-31 [2018-01-21].
11. 陳　恭. 智能合約的發展與應用 (PDF). [2019-02-20].
12. Nick Szabo -- Smart Contracts: Building Blocks for Digital Markets. www.fon.hum.uva.nl. [2017-07-29].
13. “Hyperledger,” Wikipedia, 19-Apr-2021. [Online]. Available: https://en.wikipedia.org/wiki/Hyperledger. [Accessed: 08-Jun-2021].
14. “Hyperledger Fabric,” Hyperledger, 29-Jun-2020. [Online]. Available: https://www.hyperledger.org/use/fabric. [Accessed: 08-Jun-2021].
15. “What is Hyperledger Fabric,” IBM. [Online]. Available: https://www.ibm.com/topics/hyperledger. [Accessed: 08-Jun-2021].
16. “Blockchain Vs Database - Javatpoint,” www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/blockchain-vs-database. [Accessed: 13-Jun-2021].

# Appendix

## remix code(Storage.sol)

pragma solidity ^0.4.21;

contract InfoContract {

string grad\_year = 'Null';

string stud\_id = 'Null';

string stud\_name\_zh = 'Null';

string stud\_name\_en = 'Null';

string course\_zh = 'Null';

string course\_en = 'Null';

address setID ;

bool upload = false;

function setInfo(string \_grad\_year, string \_stud\_id , string \_stud\_name\_zh, string \_stud\_name\_en, string \_course\_zh, string \_course\_en) public{

if(!upload){

grad\_year = \_grad\_year;

stud\_id = \_stud\_id;

stud\_name\_zh = \_stud\_name\_zh;

stud\_name\_en = \_stud\_name\_en;

course\_zh = \_course\_zh;

course\_en = \_course\_en;

setID = msg.sender;

upload = true;

}

}

function getInfo() public constant returns (string, string, string, string, string, string) {

return (grad\_year, stud\_id, stud\_name\_zh, stud\_name\_en, course\_zh, course\_en);

}

function change(string \_grad\_year, string \_stud\_id , string \_stud\_name\_zh, string \_stud\_name\_en, string \_course\_zh, string \_course\_en) public{

if(setID==msg.sender && upload){

grad\_year = \_grad\_year;

stud\_id = \_stud\_id;

stud\_name\_zh = \_stud\_name\_zh;

stud\_name\_en = \_stud\_name\_en;

course\_zh = \_course\_zh;

course\_en = \_course\_en;

}

}

}