Final Developer Documentation

Internship Period : 4 January 2021 - 30 April 2021

Team :

Othalia Doe-Bruce : CEO of InnovFin Consulting Inc

Punyaja Mishra : Junior Blockchain Developer (Intern)

Vanshree Mathur : Junior AI Developer (Intern)

Riley Dawson : Junior Marketing Specialist (Intern)

Ultimate Goal : Taking the demo state of the certification system to an MVP state that can be presented to potential clients.

InnovFin thrives to provide knowledge to its users on advancing and growing technology like Blockchain, Artificial Intelligence, Augmented Reality and Virtual Reality. The website is being built based on these technologies. One part of the platform is the certification system, that should provide certification to the users when they request for it after completing a course or module. This certificate details should be recorded as properties in a block for every transaction. The student details should be stored on the certificate and it should be saved as a pdf.

Start of the Internship :

Blockchain is a big and broad topic that requires a proper study and practice to be able to create a working smart contract on a network. First few weeks of the internship was spent on the online research of blockchain.

There are various blockchains right now being used, but the most utilized (after bitcoin) is Ethereum and Hyperledger. InnovFin had to decide which blockchain type would be better for the platform. I created a report on the pros and cons of the blockchain types to get a decision between Ethereum and Hyperledger Fabric. Each platform has their own tools and methods for blockchain network. Ethereum is more used for Business to Customer and generalized applications. While Hyperledger is used for Business-to-Business Transactions.

Ethereum vs Hyperledger

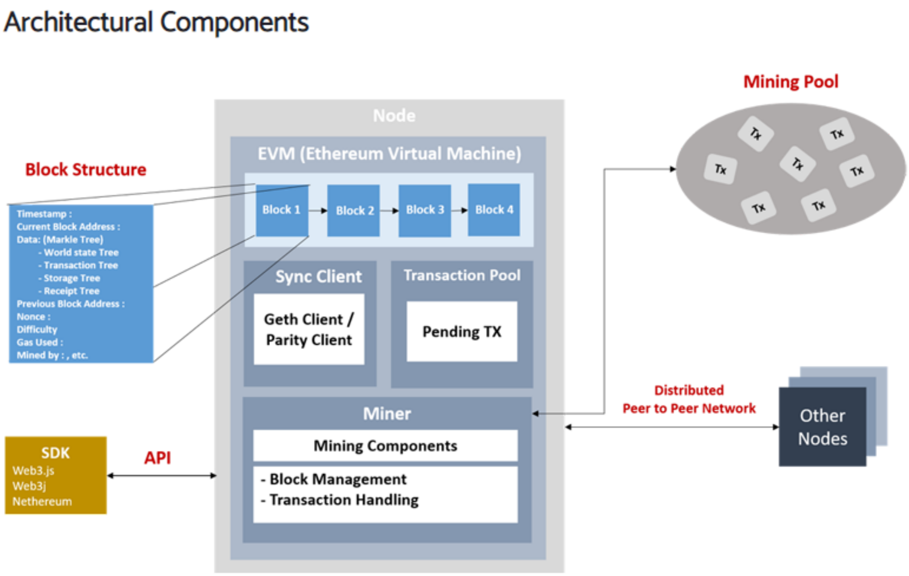
Ethereum

Pro :

1. Thinking that our platform is focused on providing educational modules and videos, courses on various upcoming emerging technologies to people all across, the transaction is Business to consumer and not within businesses. Thus, Ethereum would be a good choice for that.
2. There is a cryptocurrency Ether used, which can also be used for our network for various activities

Cons :

1. Ethereum is transparent all over the network. So, anyone using this network will be able to see each other's progress which might not be a good idea for a education platform. It is a public/private permission-less network.
2. It uses a proof-of-work algorithm, which means mining would be needed which can prove to be costly.
3. The build-in cryptocurrency value keeps changing and sometimes it can be costly
4. The transactions cost fees like fuel money, this amount keeps on altering everyday and only in upward direction.



This is an Ethereum typical architecture. I have included the source for the image. What is happening here is that each participant has a node, that connects with other nodes and a miner who mines data to ensure that the transaction is legit for the node. A node is any user or client device that is communicating with the Ethereum network. Each node contains blocks. Block is a package of data that stores transactions, hash of previous block (to keep a chain link network) and additional data like who was it mined by, fee (gas used).. Miners are the ones who add a block to the network once the transaction is confirmed. Mining Pool are just groups of miners doing their mining, confirming transactions together. SDKs are just different libraries used to interact with nodes on the network.

Hyperledger

Pros :

1. Hyperledger carries confidential transactions and would be a good choice since uses taking courses might not want any other participant to be able to see their transactions like the amount of points they earned.
2. Hyperledger uses a pluggable consensus algorithm, meaning that there is o mining required.
3. Hyperledger supports pluggable implementation of components delivery at high degrees of confidentiality, resilience, and scalability

Cons :

1. Hyperledger is a good choice for business-to-business transactions where the size of the network is limited and may not keep growing. It is a private and permissioned network.

Hyperledger Fabric Architecture

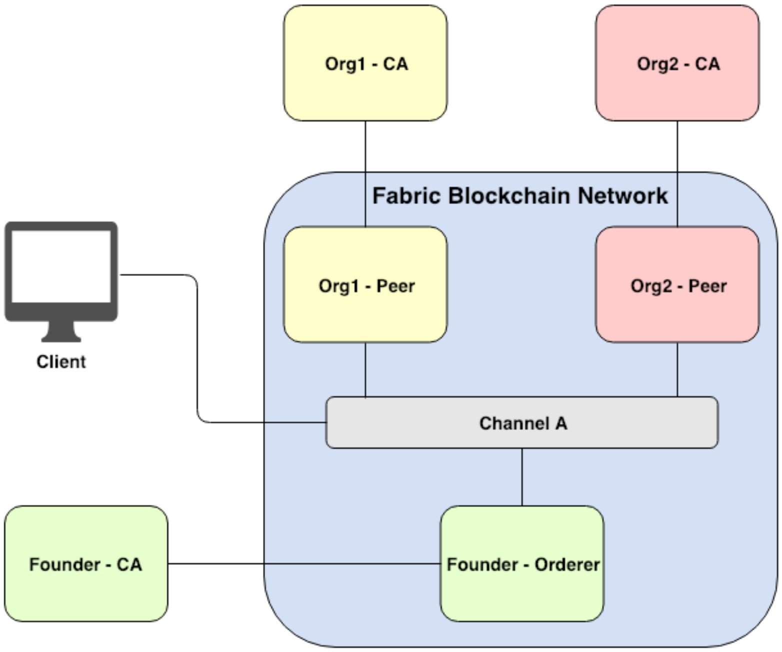


Figure 1. Simplest Fabric network with two organizations joining the same channel

This is a simple Hyperledger typical architecture. A channel is like a link that helps all the organizations to communicate with each other. Since Hyperledger is a private network, this means anyone who is not in the channel can not access any data. Peer is the node that stores all the transactions when they join the channel – ensuring confidentiality. Orderer is responsible for ordering transactions and creating new block of ordered transactions. CA is the certificate Authority for managing registrations, user enrollments and stuff – user data. Client is considered to be an application that interacts with network based on the permissions It has been specified.

Loyalty Program

For the loyalty program platform, there are a few things to be kept in mind. It would be good to have the program on **Hyperledger Fabric** for privacy purposes – any user might not want others to know how much loyalty points they have earned.

Deciding Hyperledger Fabric as the type of blockchain, the next step was answering a few questions on Hyperledger Fabric to understand how the process must go about.

Considering questions required for building efficient Loyalty Program

Here are a few questions that were answered and though about to decide on the process and make it efficient.

1. **A defined use case with the projected benefits (to you and the participants)**

Actor : User using Loyalty Program Platform

Stakeholders : InnovFin, because the blockchain (for the loyalty points) network exists because of InnovFin. Any other University offering services and/or offering their own custom points, metamask digital wallet (since it is a software required for this platform). InnovFin or University partners (including their campus stores they have links to) offering services, products and points.

Preconditions : Wants to create an account, for some users - has access to the network, Have a metamask wallet

Triggers : Anyone who decides to create an account with Innov-Edu platform (– they will automatically be assigned a wallet address and issued with points); University and other partners will have a node created to join the network; have a metamask wallet; external triggers like completing a quiz or level for a particular module; any friend or family sending their redeemed points to another account holder

System :

1. User enters the Loyalty Program platform and signs up for earning points
2. User has finished a course/module and opts for earning 10 loyalty points
3. Admins checks the respective boxes to assign 10 loyalty points to the user
4. User earns the 10 points into their metamask wallet

5. User completes a quiz or a particular level within a course module, they earn 'x' points
6. User is sent points by a friend/family member
7. User can redeem these points for various activities within the system, various services being offered by some partners
8. Administrators are notified, and they ensure the particular service or product by the partner issued the points reach the user(/student)

Goal: Provide user an access to earn points for themselves that can be redeemed later

Benefits : Loyalty Program can act as a motivation for users to use Innov-Edu platform more, for education modules, and other various activities available on the network. This may also attract new users to use the platform and enhance and grow the network further. The points can be redeemed and exchanges with friends and family, this will get more attention towards the platform and attract more users, customers that will beneficially grow the network.

2. Identified participants

It is important to identify participants.. The main participants that the network will include are :

- ❖ **Students** : The new users that will log into the platform to earn points by either completing an education module or other way
- ❖ **Admins** : The group of administrators working behind the network and completing the required steps to make sure that the students/users are getting their well-deserved points. Admins are also responsible to ensure the particular service or product by the partner reach the user(/student)
- ❖ **Other Node Administrators :** Universities and other partners assign an administrator to manage their nodes on the network
- ❖ **InnovFin platform (as a whole)** : InnovFin must also own a node and is a participant on this platform
- ❖ **Universities** : Other universities that are on the network and upload education modules. For instance, the more the students that take their module/course, they more points they earn.
- ❖ **Individual Teachers** : Users registering as a teacher and uploading their education modules and videos can also earn points. For instance, the more the students that take their module/course, they more points they earn.
- ❖ **Partners** : These are the partners offering services or products to the users of Innov-Edu platform in exchange of the redeemable points.

```
abstract participant Individual identified by Id {
  object String Id    //identification number
  private object  String u_id          //unique identification number not to be shared
  object String name          // name of the individual
  object Address address   //the HyperLedger address of the individual
}



participant University extends Individual {
  int numberOfRegisters   //the count of people who signed up for this module
  string id_group              //a string id to know if individual teacher or a University
}

public Admin extends Individual{
  string employeeId          //variable to separate admins from regular users
}
```

3. The process mapped with key events

a. An individual comes up on the Innov-edu Loyalty Program platform and browses over the Loyalty points service we have to offer : It should be ensured that the website is easy to navigate so that anyone new coming on the website can find everything easily and understand the program effortlessly.

b. The individual decides to sign up to the program. They can register into a course and then sign up for the program. When an individual signs up, they should asked if they have an existing Metamask Wallet. If they say yes, then the address will be taken from the extension. Else, if they choose “no, sign me up”, then they should be assigned a wallet. Only the public address will be with InnovFin and the private key and personal information would be only with the individual. This is done because then they can be provided with the welcoming points and automatically have an option of redeeming points.

c. Another individual who is a teacher may decide to sign up, in the same way as the student. They must know their specifications on how they can earn points, and that it is different than a student user.

d. A university and other partners need to have a node so that they can join the network and offer their services, products and also

e. The administrator gets the request for earning loyalty points (also certificate) by the Individual (the address and the individual Id which is the identification string value for individuals). The administrator issues the points to the individual either manually or automatically (later stage).

f. The Individual receives the points, and they can redeem them by clicking on the redeem points button. They can chose to use the points for any activities or other action on the network.

g. After a user redeems their points, admins should be notified. Admins will then ensure the particular service or product by the partner issued the points reach the user(/student).

4. Critical data to be exchanged

Critical Data to be exchanged :

a. An individual requesting points, the string Id value shared to the admins is critical data.
b. The individual's metamask digital wallet private key (only public address can be shared) where the points are stored is critical data and must not be shared with anyone. Also information about the amount of points earned is private to the particular student, InnovFin and whatever University issued the points.
c. The information about the amount of points redeemed and name, contact information about the individual will be private to the individual redeeming those points, University who commissioned those points, partner they are getting the product from (they know the amount of points required) and InnovFin. This will not be true for partners offering a service since they only need the email of the individual to send them the service.
d. An individual trying to redeem their points by sending it to some other user on the network, the information about the other individual should not be shared more than necessary. This will be private to the individual sending the points, individual receiving the points, University who commissioned those points and InnovFin.
e. The collection of information that the administrators need to maintain the node they are responsible for is also private and mustn't be shared with anyone.

5. Any rules to be applied in the system

There are few critical points to be taken care of which can act as rules for the system :

a. No individual should be able to look at someone else's earned points for privacy purpose
b. The private metamask wallet key of an individual should not be allowed to shared even by mistake. The id is the string value that must be shared with others for identification in case some one else decides to redeem their points and send them to other person.
c. Metamask wallet private key should be secure and not accessed by anyone else. This should be mostly coded on the user-side and not server-side to avoid any such mistake.

d. Product partner will need to know the name, address, contact information of the individual when the individual is using their redeemable points to get a product from the partner. Services being offered by the partner can be sent online via email and hence nothing except the email must be shared.
e. The data about the node being managed by administrators must not be shared

---

What are your thoughts on us moving the current Loyalty Point Platform to the use of Hyperledger?

Loyalty Program Platform - Hyperledger

Hyperledger is a private permissioned network, that means it allows confidentiality, resilience, and scalability. Hyperledger has the capacity to recover quickly from difficulties and also change in scale and size. This means, that as more user join the network and Innov-edu platform grows, Hyperledger will be easy to maintain. And a private network will allow the users to have confidentiality on the number of points they have been able to earn. Also, the transparency and anonymity outside network can be chosen by the inner organization of business, i.e., InnovFin. No consensus-based mechanism ensures that any changes will not require a unanimous agreement.

Channels in Hyperledger will ensure and help in data partitioning that will help in keeping certain data information hidden. Hyperledger fabric will allow to work in collaboration easily.

One question that may arise is that a permission-less network requires that users be granted both network and application access just to connect. However, this is a good thing since all users will be kind of authorized to join and hence protect the privacy of the network.

Over everything, Hyperledger, unlike Ethereum, do not have a constantly rising gas fee and this is very beneficial. Hence, Hyperledger seems like a good option for the Loyalty program platform.

A student should constitute a node. How do you envision students maintaining their nodes?

For student node : this will be client type node since they will be sending transaction proposal to us where transaction is asking for issuing loyalty points

Since a student is not exactly handling any developing side, there wouldn't be any hardware requirement for the student. Also, a student node will handle transactions like receiving loyalty points being issued by the Admins, which since now on Hyperledger, will not require any mining to confirm the transaction. Other than that, there are just personal data information in the node that student needs to maintain and keep with them. They need to ensure that they keep their metamask addresses safe for ensuring safety of their transactions.

How can we involve companies that will provide us with their products and or services to be exchanged on the loyalty platform? Should they also get a node?

Any companies getting involved with the network can also be provided with loyalty points – more than a student is being provided since they are providing with service? This will improve customer retention for the operation. Also, the more users use their services, the better the points they receive.

Many companies may find this as an opportunity to contribute data and services they have wanted to. By sufficient permissions, the companies can analyze the data in order to identify latest education or any other service trends among the population and hence evaluate their program's effectiveness. Also, working on Hyperledger Fabric will allow easy collaborations and stable blockchain.

They can be provided with the client node as well. This way, they can efficiently keep a count of the number of users using their services while maintaining a unique identification.

Based on your understanding of Hyperledger, how many client nodes can be issued on the network without clogging the transaction speed?

I am not a lot sure about this. I read through many sites, and it is clear that there is not exactly any limit on the number of client nodes we can have. Let's say if there is a lot of nodes, and it starts clogging the network, and the transactions start getting delayed by seconds, then we can perhaps have a clustering stage where we make a cluster of few nodes with issuing a leader node via a leader selection scheme. Also, leader is chosen through previous transaction and reputation basis, and there is a reward for leader when they do a good work in propagating a transaction. Leaders should also be updated time to time so every client node does get a chance. I am not a lot sure though.

Are we able to create channels where only client nodes share information with clients nodes? That would be student nodes sharing information with service

provider nodes?

Yes, I think we should be able to create a new channel that only allows the client nodes to enter. No ledger can pass from one channel to another, hence this would be also maintain a good privacy. This isolation of clients and ledger data, by channel, allows the client – network members – to have transactions on the same Blockchain network. For Student nodes sharing information with service provider nodes, there could be a channel with the leadernode as the service provider?

After thinking about all questions, Hyperledger Fabric looked like a good option to be chosen as a blockchain type. However, as we know, every transaction requires gas fee and Hyperledger fabric does not allow a good test network. Hence, it was decided that for now the platform will continue to run on the Ethereum Kovan Test Network – where the platform had been built till now.

Developer Tools

Next step is to decide what tools to be used by the developer to build the platform using. There are various tools that can be used for Ethereum and Hyperledger Fabric. Each blockchain type has their own set of tools that work well for them.

Ethereum

1. Truffle
2. Ganache
3. Drizzle
4. Remix IDE
5. Solidity
6. Test RPC
7. Mist

Hyperledger

1. Golang
2. JavaScript
3. Nodejs
4. Java SDKs
5. Docker
6. Metamask

Tech Profile of existing platform

Before moving further, I was asked to find the platforms this site is already hosted on. And I made a tech profile of the website for the main page and the verify certificate page.

WordPress Plugins :

1. **Akismet :** spam filtering service that filters spam from comments, trackbacks, and contact form messages. The filter works by combining information about spam captured on all participating sites, and then using those spam rules to block future spam.
2. **Contact Form 7 :** This plugin is simple but flexible – it manages multiple contact form, plus you can customize the form and the mail contents flexibly with simple markup – CAPTCHA is included in this
3. **Jetpack by WordPress.com** : Plugin brings many of the most powerful features available on WordPress.com to self-hosted WordPress sites
4. **All in One SEO Pack** : This plugin allows to easily optimize the WordPress website for search engines and social media platform
5. **Google Analytics for WordPress** : Adds tracking ID to every page of site automatically – automatic tracking of outbound clicks and pageviews
6. **LearnPress Plugin :** Allows for multiple instructors, where two or three manage one course

API :

1. **Google Font API :** The Google Font API helps add any web fonts to any web page

Frameworks :

1. **PHP; PHP7 :** language used for web development and can be embedded into HTML

Mobile :

1. **Iphone/ Mobile Compatible :** Website contains code that allows the page to support iphone and android or any other Mobile Content.
2. **ViewPort Meta :** There is viewport meta tag which means the content may be optimized for mobile content – also plugin All in One SEO helps in optimization.

Web Hosting Providers :

1. **Amazon – AWS –** Amazon Web Services : the most on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis

Email Hosting Providers :

1. **SPF :** Sender Policy Framework is an open standard specifying a technical method to prevent sender address forgery
2. **Namecheap :** This Email hosting provider has good website security and privacy for hosting custom domains for emails.

Name Server :

1. **Namecheap DNS :** This DNS service provides name to IP address mapping for devices connected to Internet.

Web Servers :

Nginx : HTTP server and mail proxy server

Verify Certificate Page

Frameworks :

Express.js : JavaScript framework used to build web application in Node,js

Webservers :

Same webserver as the main Innov-edu site.

Tools :

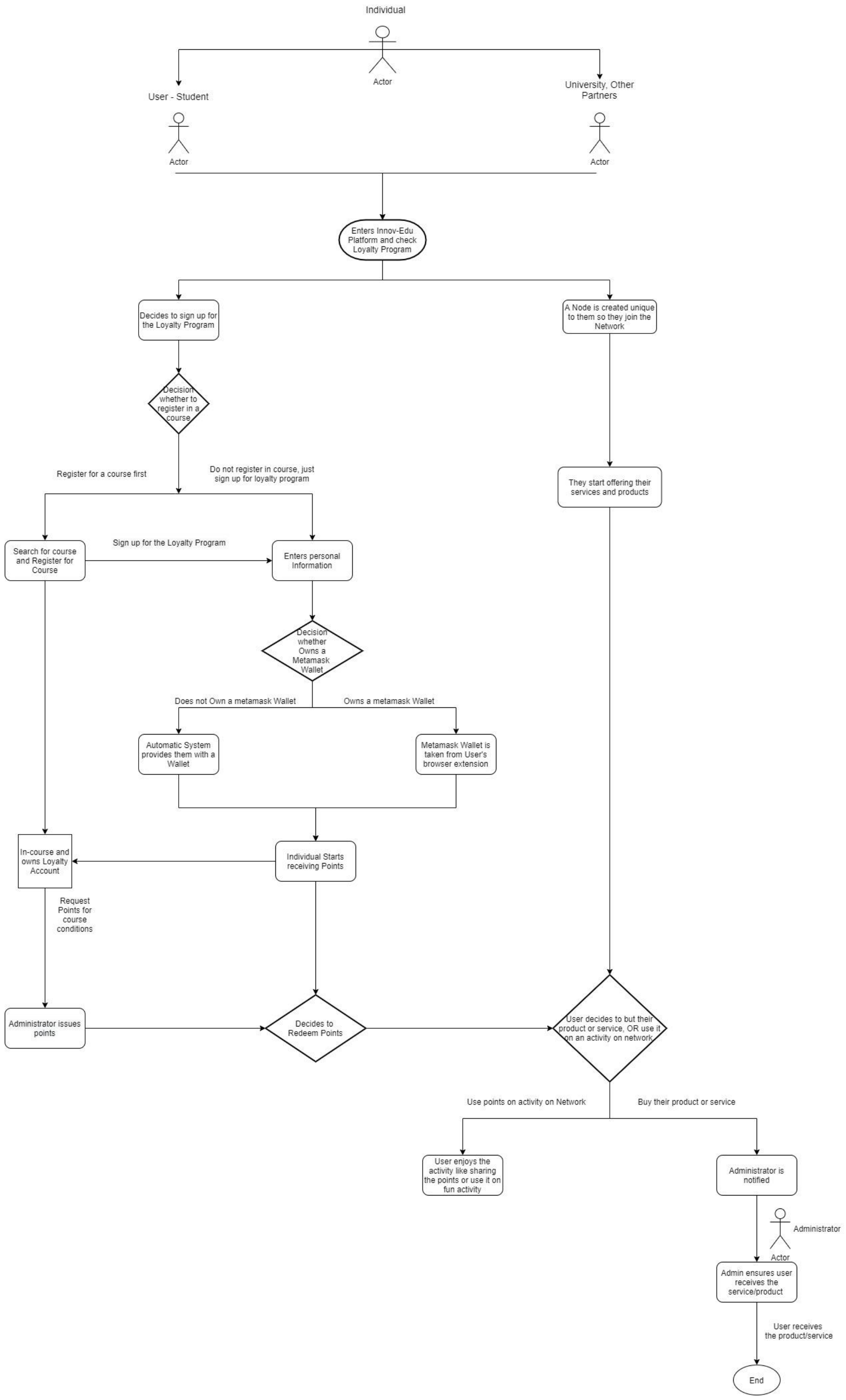
1. **HTML5 DocType**
2. **CSS**
3. **Twitter Bootstrap :** Designed to kickstart development of webapps and sites
4. **Sub Resource Integrity :** SRI is a W3C specification that allows web developers to ensure that resources hosted on third-party servers have not been tampered with

Content Delivery Network :

1. **Content Delivery Network** : some of the site contents are stored on a content delivery network – proxy server
2. **StackPath BootstrapCDN** : Their CDN System – enables to load CSS, JS and images remotely from the servers

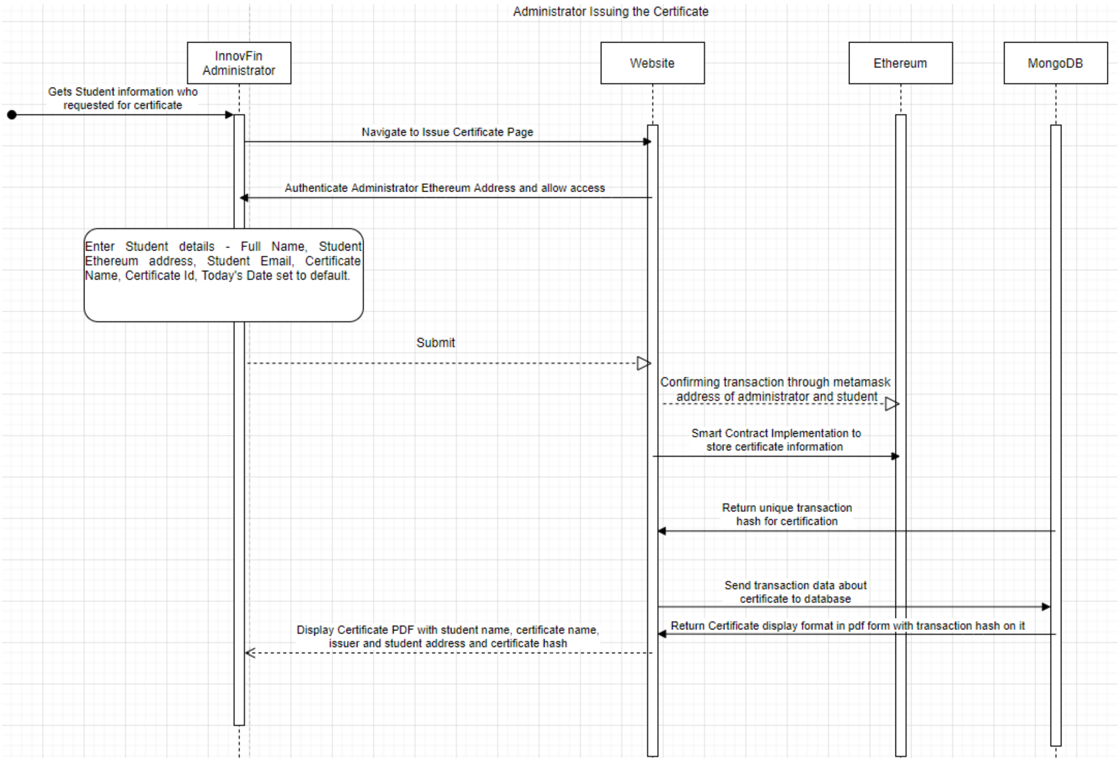
Moving on to the process of creating the certification system

Project Mapping Key events

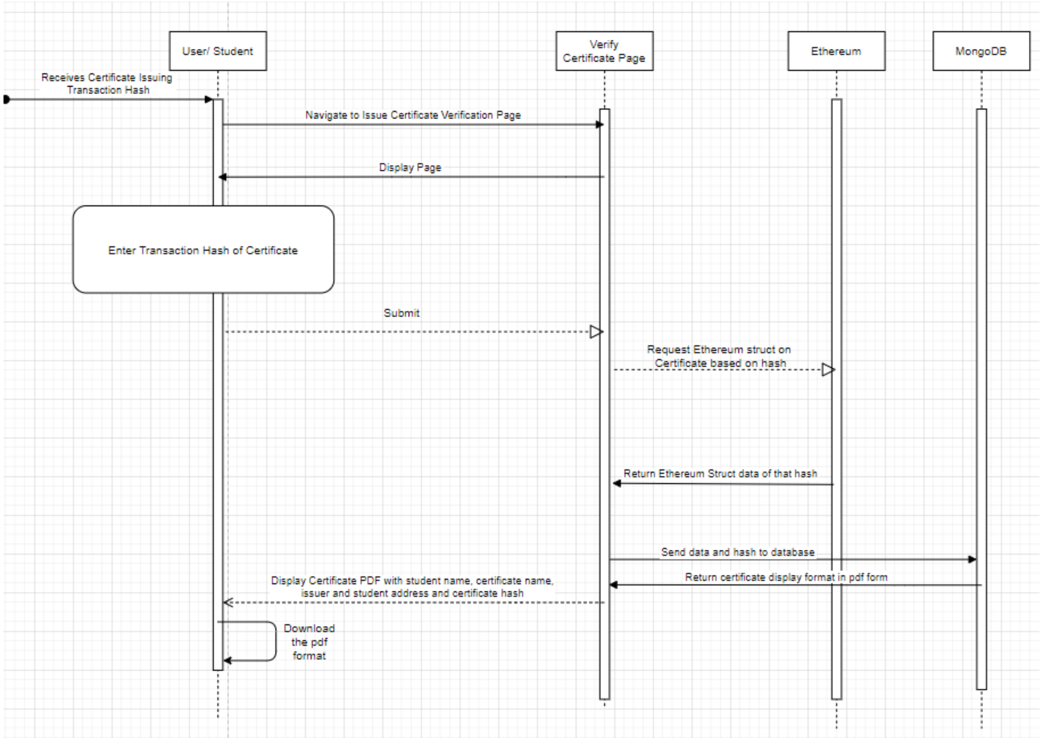

Individual
Actor
User - Student
Actor
University, Other Partners
Actor
Enters Innov-Edu Platform and check Loyalty Program
Decides to sign up for the Loyalty Program
A Node is created unique to them so they join the Network
Decision whether to register in a course
Register for a course first
Do not register in course, just sign up for loyalty program
They start offering their services and products
Search for course and Register for Course
Sign up for the Loyalty Program
Enters personal Information
Decision whether Owns a Metamask Wallet
Does not Own a metamask Wallet
Owns a metamask Wallet
Automatic System provides them with a Wallet
Metamask Wallet is taken from User's browser extension
In-course and owns Loyalty Account
Individual Starts receiving Points
Request Points for course conditions
Administrator issues points
Decides to Redeem Points
User decides to but their product or service, OR use it on an activity on network
Use points on activity on Network
Buy their product or service
User enjoys the activity like sharing the points or use it on fun activity
Administrator is notified
Administrator
Actor
Admin ensures user receives the service/product
User receives the product/service
End

Sequence Diagram of events to understand the flow of process better in a technical way

Administrator gets the request for issuing a Certificate


Administrator Issuing the Certificate
InnovFin Administrator
Website
Ethereum
MongoDB
Gets Student information who requested for certificate
Navigate to Issue Certificate Page
Authenticate Administrator Ethereum Address and allow access
Enter Student details - Full Name, Student Ethereum address, Student Email, Certificate Name, Certificate Id, Today's Date set to default.
Submit
Confirming transaction through metamask address of administrator and student
Smart Contract Implementation to store certificate information
Return unique transaction hash for certification
Send transaction data about certificate to database
Return Certificate display format in pdf form with transaction hash on it
Display Certificate PDF with student name, certificate name, issuer and student address and certificate hash

Student receives hash and goes to verification of certificate


User/ Student
Verify Certificate Page
Ethereum
MongoDB
Receives Certificate Issuing Transaction Hash
Navigate to Issue Certificate Verification Page
Display Page
Enter Transaction Hash of Certificate
Submit
Request Ethereum struct on Certificate based on hash
Return Ethereum Struct data of that hash
Send data and hash to database
Return certificate display format in pdf form
Display Certificate PDF with student name, certificate name, issuer and student address and certificate hash
Download the pdf format

Designing a plan to make the demo to reach the MVP state

For a product or demo to reach a MVP state means making it to be a 'minimum viable product'. This version of our product will include all the features necessary to release the product into the market by solving the core problem for a set of users. This plan will allow us to provide immediate value, while minimizing development costs. To design a plan to bring the demo to MVP state, we need to follow and understand the valuable steps of : Understanding the Business Needs, Finding the Opportunities, and Deciding what features to build (Prioritization Matrix).

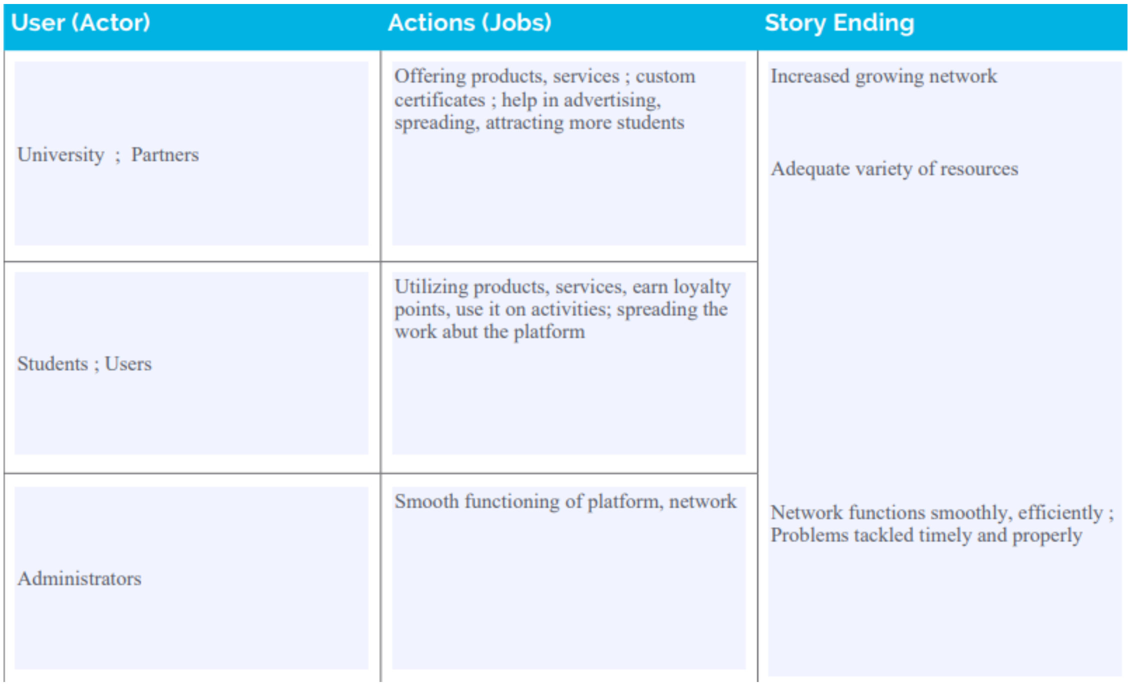
Understanding the Business Needs

A product is fit to acquire a place in the market once it has a good UX/UI with adequate feature set and value proposition and also a good foundation of knowledge on the target customers.

Knowing the long-term goal gives a good start to the plan. Our long-term plan includes a well functioning platform with educational services, diverse products, activities and services for users to enjoy and also creating a good space for knowledge on the upcoming advanced technologies. Understanding user-personas is important. This can be done by recognizing the needs of the customers. The expectations of users include being provided with a good educational service and an easy to navigate platform that displays adequate information on the site about the various features and activities the site has to offer. Some motivations for attracting a good customer base is the loyalty program page, various activities and interesting courses and modules. There will also be a few pain points that customers might face. For instance, no knowledge about the advanced tech and hence probably facing trouble in understanding either steps or modules easily. The success criteria can be measures by an increased number of users logged in, growing blockchain network, escalating ties-up with universities and other partners who are ready to offer products and services.

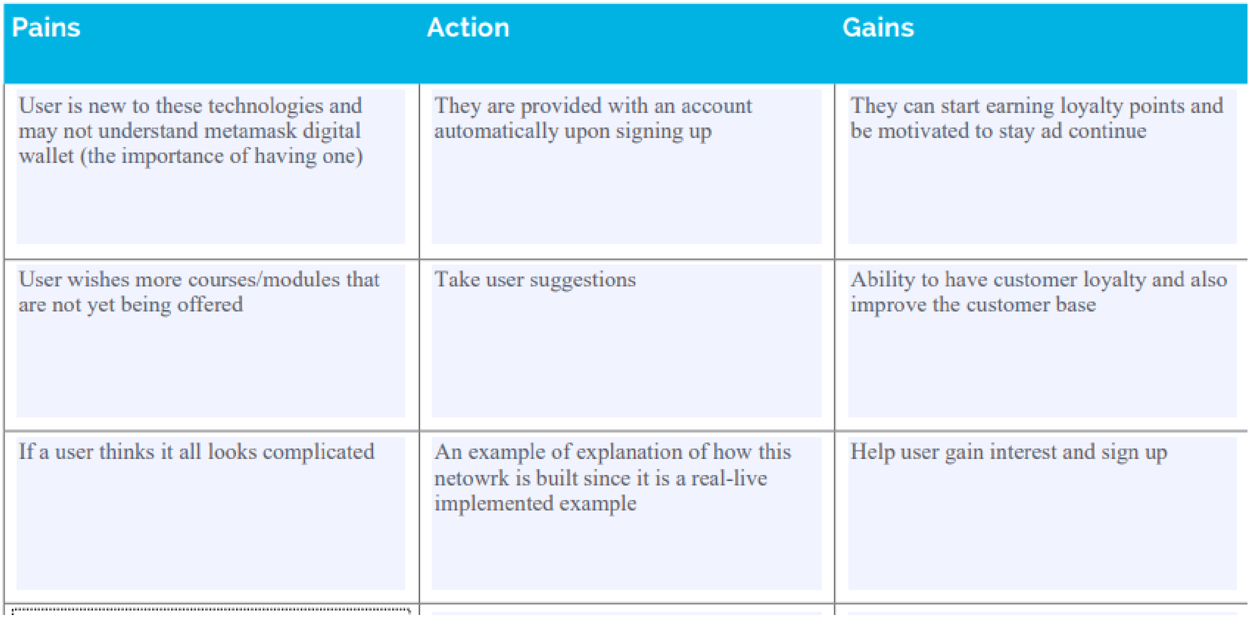
Find the opportunities

An opportunity can be found by listing down the users, their jobs and how we want this process to end.



User (Actor)	Actions (Jobs)	Story Ending
University ; Partners	Offering products, services ; custom certificates ; help in advertising, spreading, attracting more students	Increased growing network Adequate variety of resources
Students ; Users	Utilizing products, services, earn loyalty points, use it on activities; spreading the work abut the platform	
Administrators	Smooth functioning of platform, network	Network functions smoothly, efficiently ; Problems tackled timely and properly

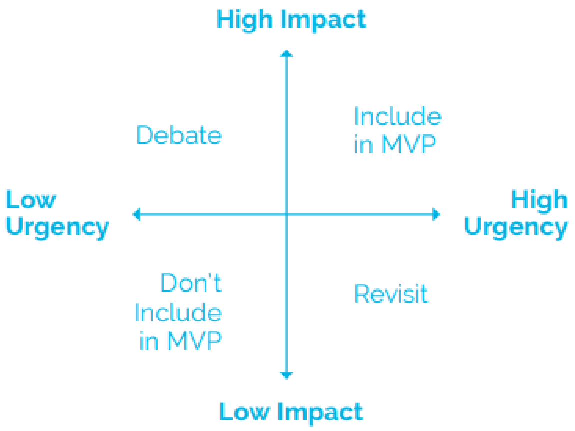
Similarly, understanding pains and what actions to take for overcoming those and entrapping a gain is important.



Pains	Action	Gains
User is new to these technologies and may not understand metamask digital wallet (the importance of having one)	They are provided with an account automatically upon signing up	They can start earning loyalty points and be motivated to stay ad continue
User wishes more courses/modules that are not yet being offered	Take user suggestions	Ability to have customer loyalty and also improve the customer base
If a user thinks it all looks complicated	An example of explanation of how this netowrk is built since it is a real-live implemented example	Help user gain interest and sign up

Those tables help us realize opportunities like how we can gain audience trust and loyalty, grow our network and also improve the platform.

Prioritization Matrix :



This prioritization matrix finally helps in building an efficient MVP.

For instance, finding Universities and partners that are ready to offer products and services are impactful but not urgent and hence they are debatable to be included in MVP. On the other hand, having a few modules from either InnovFin team is impactful and urgent and hence will be needed in MVP. Automated issuing and assigning of certification again debatable. Now, a chat feature for helping users guide through the website may have low impact and is definitely not an urgent requirement, hence it can be avoided from MVP.

Few more points to keep in mind :

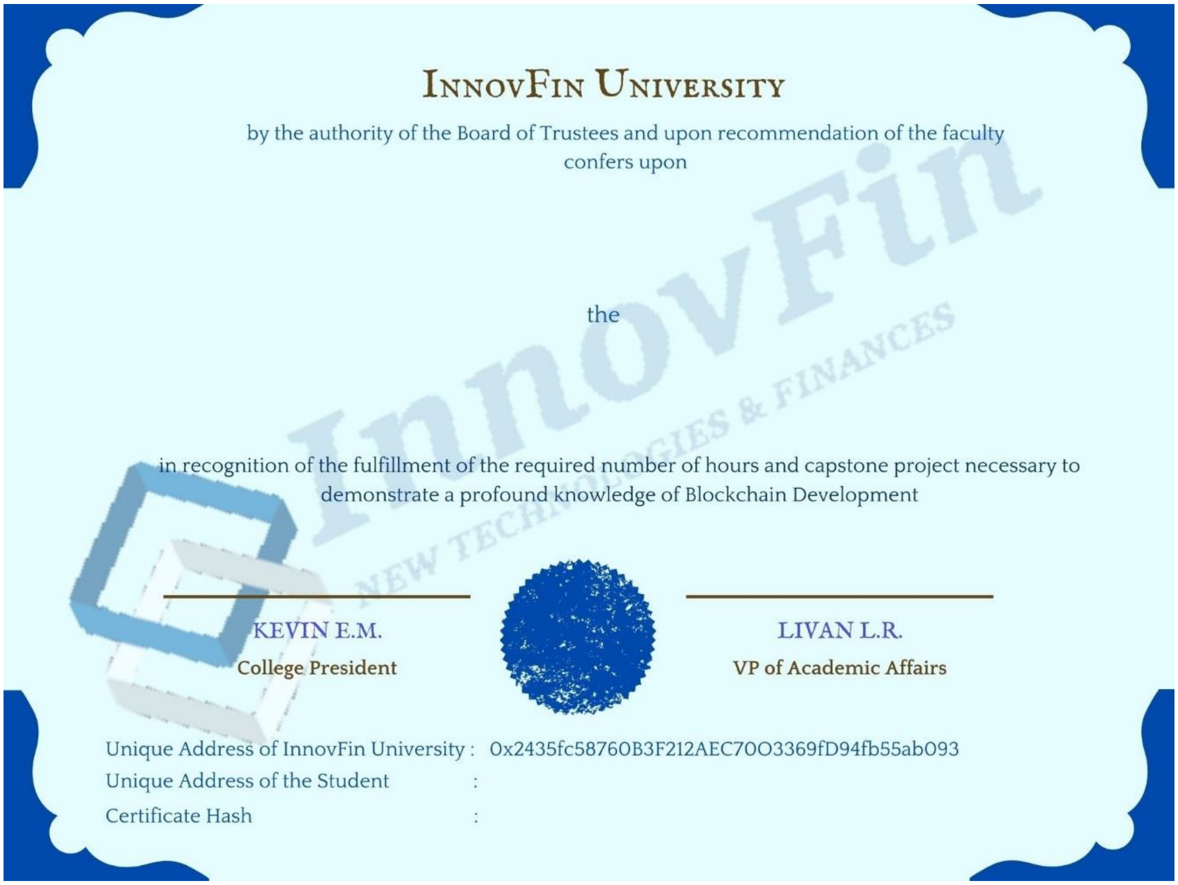
The main page of Innov-Edu platform should be implemented efficiently. An overall good design would include consistency of colors and design, while having easy navigation. A menu bar with all options self-descriptive. Information blocks about the program and platform and few extra articles for explanation about the topic.

Certification Program with a good interface and explanatory blanks so user knows what to enter and where.

The platform should allow a user to sign up with or without metamask digital wallet. They can access the courses and modules on the platform and earn points which they can redeem for various methods that they can choose.

Designing the certificate

For the certification system, we need a good template of certificate. InnovFin had a design of certificate built by the previous developer (one before me). For creating the website that will print a certificate in the pdf form with the student/user’s details requesting the certificate, we need a plain certificate with nothing written on it and in a good pixel format. Using Gimp, and referring to a few designing templates of a certificate, I created this certificate from scratch.



INNOVFIN UNIVERSITY

by the authority of the Board of Trustees and upon recommendation of the faculty
confers upon

the

in recognition of the fulfillment of the required number of hours and capstone project necessary to
demonstrate a profound knowledge of Blockchain Development

KEVIN E.M.
College President

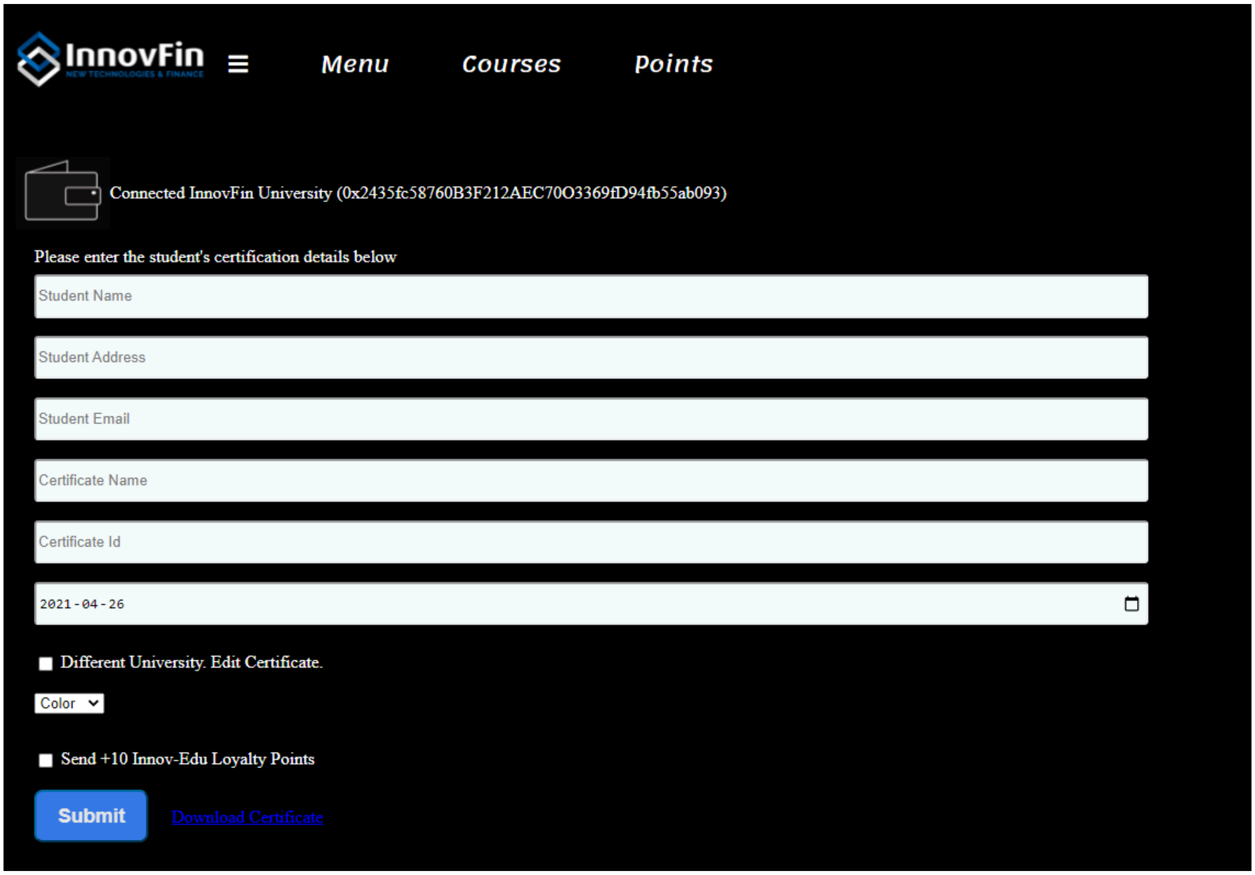
LIVAN L.R.
VP of Academic Affairs

Unique Address of InnovFin University : 0x2435fc58760B3F212AEC70O3369fD94fb55ab093
Unique Address of the Student :
Certificate Hash :

Front-end Development

Now it was time to have a working front-end of the website for certification page. This included a webpage that will take the student's information.

Using appropriate architecture, I started building the site from the scratch. I used **HTML** and **CSS** for designing and **JavaScript** for applying the functions to the fields on the webpage.



Connected InnovFin University (0x2435fc58760B3F212AEC70O3369fD94fb55ab093)

Please enter the student's certification details below

Student Name

Student Address

Student Email

Certificate Name

Certificate Id

2021-04-26

□ Different University. Edit Certificate.

Color

□ Send +10 Innov-Edu Loyalty Points

The webpage does not have any problems, however, requires some work at the later stage of the project as development happens. This webpage is designed for basic functionality. Since project is at a development level right now, all the courses and modules are being offered by InnovFin and hence InnovFin administrators will be the one providing the certificate. There will be manual entry of the student's details and the from address will be the InnovFin address. However, at later stage, the idea is to make the system automated. That is, if a user requests for a certificate, this process should automatically happen (take student details and send it to the certificate and complete the blockchain transaction without the need of any manual entry from any admin).

Also, later on InnovFin plans to partner up with various Universities that wishes to provide their own modules and courses. They get their own nodes on the network and they can earn token for each new student. This is still an idea and does not contain firm details. However, there is the idea that the university should be allowed to change the color of the certificate if they wish to. And this time the from address should be the public address of the University and not InnovFin to record perfect transaction. This means that the InnovFin address written at the top, should not be printed.

Backend Development

Front-end designing and development was easy. Now the task was the important of all, the backend of the system. This means linking the JavaScript Code to the blockchain network.

Every block chain has a smart contract. A smart contract is a set of rules that decides what should happen at every transaction on the network. To develop a smart contract there are various tools and software available. I decided to start with **Remix IDE**. Remix is a well-known software that is used to create smart contracts in language solidity. Remix does not take reference to outer repositories though and I did not know this at the time I started writing this smart contract. Writing a smart contract was not easy since I had never written it before. I tried looking for a few good resources to refer to so as to get an idea and be able to develop a good and efficient working smart contract. Github and Etherscan is usually a good pool of resources for every project. There was a very similar processing smart contract on https://etherscan.io/. However, it was in the language Python. Since, I do not know and understand Python yet, I could not take that reference and use it for my smart contract. The address I used to access that transaction was given to me by Othalia and since it is private to her, I will not include the address here.

I had seen a few codes in Solidity while researching about blockchain and hence decided to move forward with the language for my contract. I am using the open zeppelin library in my smart contract. I import the ERC721.sol and Ownable.sol from open zeppelin. I had started writing the smart contract on Remix IDE. And I was using Remix IDE which does not link to outside libraries. I did know this at that point and hence when I tried compiling, it started showing errors on the code lines with import “ ”. I tried various different ways to denote the path, using node modules, using github repository URL but none worked. After a while of changing paths, I just copied the website URL and that stopped showing error and let me compile the code. I copied the ABI after compiling and saved it. That’s how I got my smart contract done.

Implementation of the smart contract

Implementing the smart contract was not easy. There are many resources with various ways to implement a smart contract. Web3js is what is used to connect the front-end to the backend smart contract and the Ethereum test network. The tools that we use to get that done is what differs.

Ganache is a tool that will give test private keys with 100 ethers each which the user can use to test on the network. It can be used to run tests, execute commands, and inspect states while controlling how the chain operates. Truffle is basically a development environment where we can easily develop a smart contract with its built in testing framework, smart contract compilation, deployment, interative console and many more features.

I decided to use Truffle, Ganache, nodejs and web3js for implementing the smart contract.

https://youtu.be/jUpyJhnXRFE : I started using this tutorial videos as my resource. The videos are great for learning and understanding, however something was not running right. I was unable to connect the test network to my website and I was not sure what the problem was. I decided to change my methods.

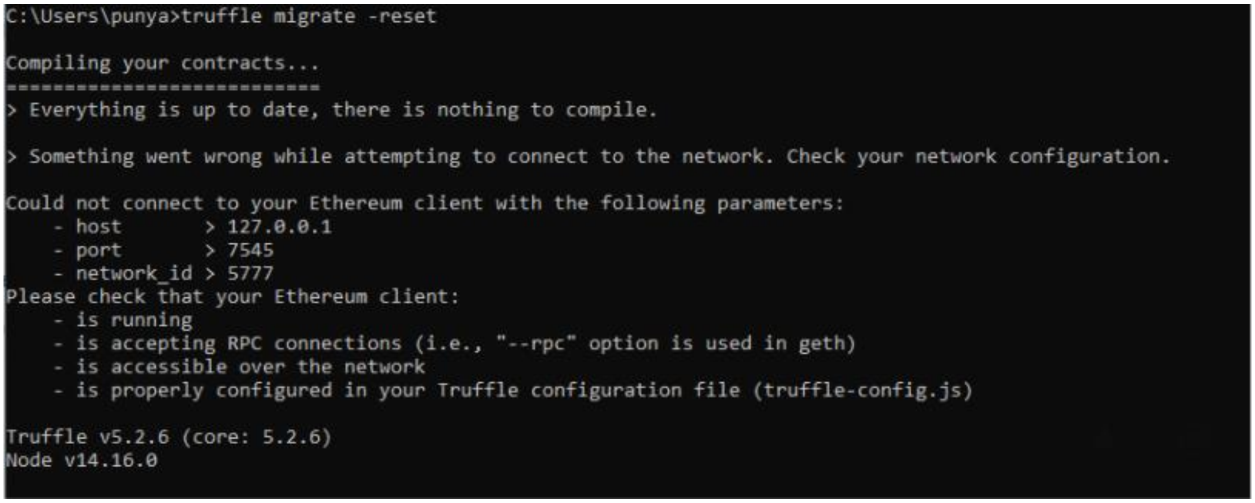
Using command terminal, I had to install npm, truffle, ganache and then use nodejs for web3js. However, it would not let me install npm and hence I could not install anything else. The entire day my terminal was unable to connect and one shift was a waste because of that.

Finally once my terminal was up again, I installed npm, truffle, ganache and I started using these links as my resources :

https://www.programmersought.com/article/64506931409/

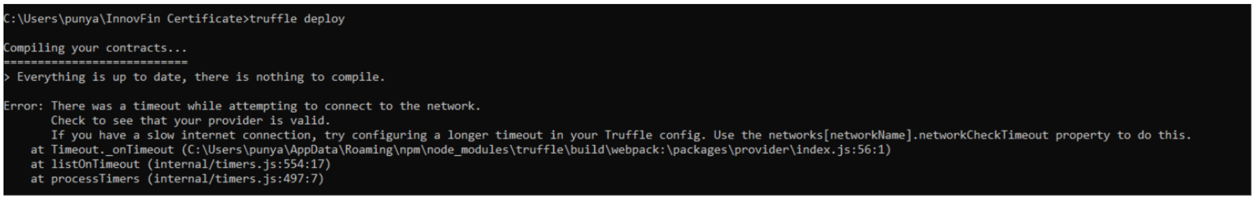
https://medium.com/coinmonks/5-minute-guide-to-deploying-smart-contracts-with-truffle-and-ropsten-b3e30d5ee1e

But I kept getting errors. It started with compiler version error which I fixed by changing the truffle-config file and adding the line specifying the solidity version.



```
C:\Users\punya>truffle migrate -reset

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.

> Something went wrong while attempting to connect to the network. Check your network configuration.

Could not connect to your Ethereum client with the following parameters:
    - host       > 127.0.0.1
    - port       > 7545
    - network_id > 5777
Please check that your Ethereum client:
    - is running
    - is accepting RPC connections (i.e., "--rpc" option is used in geth)
    - is accessible over the network
    - is properly configured in your Truffle configuration file (truffle-config.js)

Truffle v5.2.6 (core: 5.2.6)
Node v14.16.0
```

Next error was



```
C:\Users\punya\InnovFin Certificate>truffle deploy

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.

Error: There was a timeout while attempting to connect to the network.
       Check to see that your provider is valid.
       If you have a slow internet connection, try configuring a longer timeout in your Truffle config. Use the networks[networkName].networkCheckTimeout property to do this.
    at Timeout._onTimeout (C:\Users\punya\AppData\Roaming\npm\node_modules\truffle\build\webpack:\packages\provider\index.js:56:1)
    at listOnTimeout (internal/timers.js:554:17)
    at processTimers (internal/timers.js:497:7)
```

I figured, I had to just wait since again there was a network issue, and my terminal was not connecting.

Finally, when I stopped working on this project, I was having error on truffle.

'{"errors":[{"component":"general","formattedMessage":"* Line 1, Column 1\\n Syntax error: value, object or array expected.\\n* Line 1, Column 2\\n Extra non-whitespace after JSON value.\\n","message":"* Line 1, Column 1\\n Syntax error: value, object or array expected.\\n* Line 1, Column 2\\n Extra non-whitespace after JSON value.\\n","severity":"error","type":"JSONError"}]}'

I could not find any way or even a useful resource to help me our with this error, and this is where I finally got stuck.

I have been using Truffle, Ganache, Web3js.

I am assuming to get this smart contract implemented, the smart contract needs to be looked at again, compiled, get the ABI and bytecode of it and then implementation process should be started again.

End.