Minor Project Report

On

E-TRANSCRIPT MANAGEMENT SYSTEM USING BLOCKCHAIN

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
In
Computer Science and Engineering

Submitted By:

VRISHTI GUPTA
1905064
PRIYANKA JHAMB
1905379
AMANJOT SINGH
2004691

Guru Nanak Dev Engineering College

Ludhiana - 141006

Abstract

The transcript result processing and transfer is been a problem for the administrative inmost universities relating to the fact that information are been misplaced, the staff could spend a lot of time trying to locate a student's record and the delay in transfer of the documents. Over the years, record keeping was done manually, keeping paper-based documents of all students' academic information. This system however has proven inefficient because it could not cater for unforeseen circumstances like fire accident, house demolishing or even the most obvious one - the student's record going missing. Researchers have come up with more effective ways of solving the paper problem, which led to several web-based student information management systems. However, these researchers did not fully eradicate the paper problem because the process still required an initial printing, then scanning of academic records which served as inputs into the system. Also, there was an increase in the occurrence of service unavailability because all storage and processes came from the same location, making the server vulnerable to unforeseen circumstances like power outage, loss of network connection and system malfunction.

In view of this problem an automated transcript system was developed using Blockchain, IPFS, python, Flask. The new system developed has quickened the processing and transfer of transcripts and also reduce loss of data. Since it is a pilot implementation, much more features can be added at large scale.

Acknowlegement

We are highly grateful to **Dr Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC)**, Ludhiana, for providing this opportunity to carry out the minor project work at making 'Deployment of a web server on MS Azure Cloud Platform .

The constant guidance and encouragement received from **Dr. Parminder Singh H.O.D. CSE department, GNDEC** Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to **Asst. Prof. Manpreet Kaur Mand**, without his wise counsel and able guidance, it would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of the Computer science and engineering department of GNDEC for their intellectual support throughout the course of this work.

Finally, We are indebted to all whosoever have contributed in this report work.

Amanjot Singh
Vrishti Gupta
Priyanka Jhamb

List of Figures

List of Tables

Contents

1 Introduction

1.1 Introduction to Project

e-Transcript Management System Using Blockchain is a system which will be used to manage transcripts and store it securely. In the university context, a lot of efforts and costs are put into managing records for all educational transactions occurring since the first-time students submit their papers and register courses for the first semester until they graduate. In this context, security and data integrity challenges can be identified. The proposed model adds more security via the use of hashing and data readily available with decentralized data storage.

1.2 Project Category

- Application or System Development
- Network or System Administration
- Block-chain

1.3 Objectives

This project will help in clarifying issues and benefits to academic institutions and third-parties when processing and viewing transcripts of students in a block chain.
The main objectives of this project is:

1. To create a blockchain structure to store transcripts.
2. To design a user interface to access the stored transcripts.
3. To deploy and test the proposed system against existing system.

1.4 Problem Formulation

Based on the analysis of the existing transcript management, it showed problems like:

- Centralized record keeping where fault tolerance depends on a single cloud provider.
- Record misplacement.
- Going back to multiple sources for validating data.
- High time consumption.
- Non-cost effective system.

1.5 Identification of Need

Because of the problems inherent in the existing system in the transcript system, the need for computerization becomes imperatives. These are listed below as follows:

- Record and reports of students will be easily retrieved with increased data security.
- There will be reduction in the amount of resources, which in turn will lower the cost of service, since information will be stored with reduced data Redundancy.
- School personnel can attend to students without being over worked.
- There will be reduction in time used in retrieval of student files.
- Reduction in bulkiness of files and record.
- It will make available the storage room that was used for storage of files.
- Its facilities easy transaction without delay.

1.6 Existing System

The existing system of transcript management system involves lots of paper work. The system involves that a student will only get his or her transcript at the end of the program sent to school, organizations that wants to know the students performances in school leading them to know area of strength and weaknesses. Anyone that needs a transcript would have to link his/her way to the university and provide the school authorities with appropriate details before the transcript can be processed during this process lots of error might occur like loss of information.

1.7 Proposed System

The implementation of blockchain in the education sector provides a new horizon for set of non-functional requirements including but not limited to: security, immutability, independence from the institution, immutability of official records and certificates. In addition, total trust in the accuracy and infallibility are all gathered in the decentralized ledgers of blockchain.
By using blockchain technology it is entirely feasible that requesting parties will be provided the requested transcript the same day. When blockchain technology is used in the issue of certificates, there is an opportunity to not only just verify credentials without an intermediary, but also to enrich and add value to the already existent digital certification ecosystem.

1.8 Unique Features of the System

This project will help in clarifying issues and benefits to academic institutions and third-parties when processing and viewing transcripts of students in a block chain.

1. **Secured Data:** One of the objective of Transcript Management system is data security which can be achieved by using Block chain technology as it provides network users with decentralized data storage, where data is stored in blocks and is secured by using some hash value(SHA-256).

2. **Time Efficient:** Another objective of Transcript Management system is verification of transcripts in less time. Every person would have their own verifiable learning history and credentials that could be accessed from virtually any device which is an time effective solution.

3. **Immutability:** Transcripts are immutable as if one tries to change file, hash of file will be changed and will effect whole blockchain.

4. **Visibility:** The data will be visible to only authenticated persons. As certificates uploaded on blockchain, so that certificates will be visible to the person even if any failure comes on one end as this is decentralised system.

2 Requirement Analysis and System Specification

2.1 Feasibility study

- **Technical Feasibility:** It is technical feasible because user does not need to know anything technical as the interface is very handy. We have created a User Interface (UI) that allows a student to request a transcript from a particular educational institution, typically a university.
- **Economic Feasibility:** Very less time and resources are used for the verification of certificates using blockchain. We are storing hash value(SHA256) as data in the bloackchain, considering very less resources. Moreover, Data is decentralized so no load on single PC.
- **Operational Feasibility:** This project can be proven helpful for many requirements like verifiable certificates, easily credit transferable, immutability of records of official papers, certificates and educational organisations. It is operational like any educational organisation can use it. Students will not have to wait for many days to have the needful documents from the college or university. They will get a very good facility from the college because college will be using transcript management system using blockchain.

2.2 Software Requirement Specification Document

- **Data Requirement:** The transcript data of students along with their additional details were required to be stored in blocks and to observe the data through repeated careful listening.
- **Functional Requirement:** Blocks check, admin add kr ske
- **Performance Requirement:**
  - **Performance Requirement:** IPFS Daemon should be on on the server.
  - **Software Requirements:** The following specifications are required:
    - **Operating system:** The Certified distribution of Windows, or Linux or MacOS.
    - **Front End:** With the aid of Python, a programming language.
    - **Back End:** With the aid of Flask and SQL
    - **Tools:** IPFS
  - **Hardware Requirements:** The following specifications are required:
    - 64-bit CPU (Intel / AMD architecture) (At least Dual core processor)
    - 4 GB RAM
    - At least 5 GB free disk space
- **Dependability Requirement:** System depends on IPFS server, which is a decentralized storage and data can be easily shared to all peers, to be running,
- **Maintainability requirement:** This software will be easy to maintain as, using blockchain for transcript Management, data will be decentralized and will be cost efficient.

- **Security requirement:** System security depends on decentralized data management.
- **Look and feel requirement:** E-transcript management system using blockchain is an efficient transcript management system with an easy to use interface where admin can upload transcripts and user/student can access their transcripts with all security.

2.3 Validation

The software has been validated and it was found that our pilot implementation is working well.

2.4 Expected hurdles

Implementing pilot implementation to the main implementation of the project might become challenging. Hardware requirements may be increased if we implement on the large scale.

2.5 SDLC model

1. **Planning and Requirement Analysis:** Requirement analysis is the most important and fundamental stage in SDLC. The information that we got from requirement analysis used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas. Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage
2. **Defining Requirements:** Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer.
3. **Designing the Product Architecture:** The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details.
4. **Building or Developing the Product:** In this stage of SDLC, the actual development starts and the product is built. The programming code is generated during this stage. Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code.
5. **Testing the Product:** This stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.
6. **Deployment in the Market and Maintenance:** Once the product is tested and ready to be deployed it is released formally in the appropriate market.

3 System Design

3.1 Design Approach

We have used both Function oriented and Object oriented Approach in our project. As with function oriented we can have easy flow of information and in object oriented we can use concepts like information hiding, benefits of constructor and destructor etc.

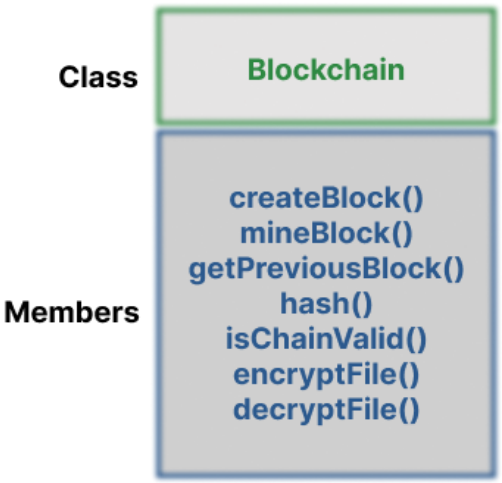


Figure 1: Design Approach

Design approach includes object oriented design as well as function oriented design where we have created **blockchain** class where data members are:

- Create Block
- Mine Block
- Get Previous Block
- Hash
- Is chain Valid
- Encrypt File
- Decrypt file

Functions included are:

- login(): Login function which is used to redirect to the user to login page.
- logout(): Student or Admin can logout using this function
- AddUser(): Admin can add user by using this function through the interface.
- Dashboard(): The main page is our Dashboard page which is used to see the certificates of particular user,
- getchain(): One can use to see the block chain that has been made till time.
- isvalid(): It will return True if the chain is valid else False.

- uploadfile(): It is used to upload the file to IPFS.
- authoritycheck(): It is used to check the authority whether a user is admin or student.

3.2 System Design

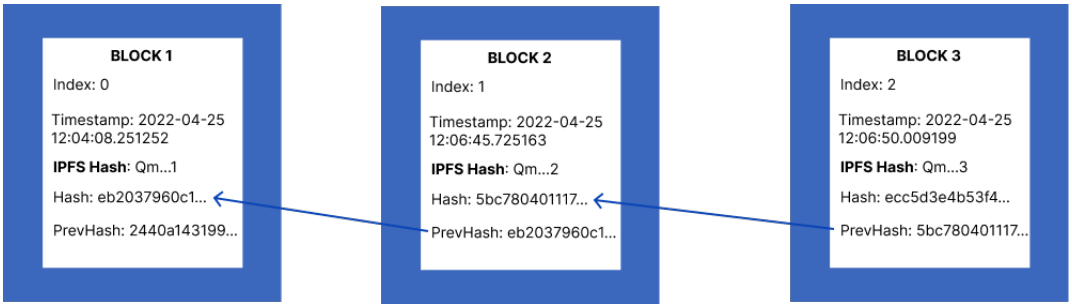


Figure 2: Blockchain Design

Blockchain is how the data is structured. A blockchain collects information together in groups, known as blocks, that hold sets of information. Blocks have certain storage capacities and, when filled, are closed and linked to the previously filled block, forming a chain of data known as the blockchain. All new information that follows that freshly added block is compiled into a newly formed block that will then also be added to the chain once filled.

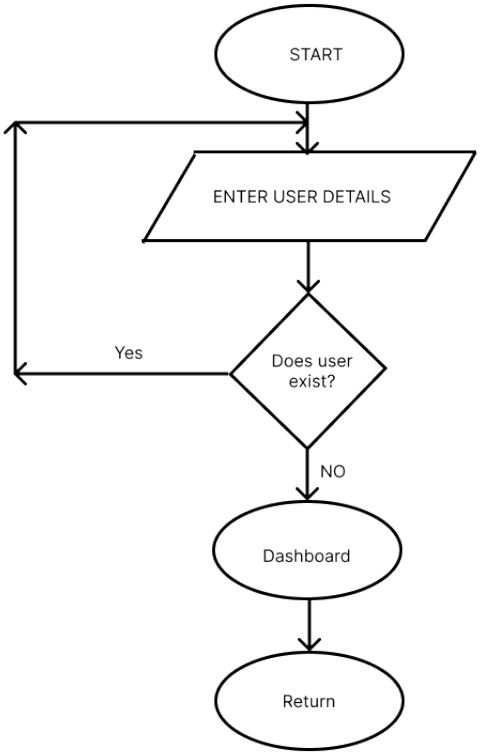


Figure 3: Add User Page

In add user page, admin adds user details of the student and if the student already exist then re-enter the correct student details and if it does not exist then student will be added.

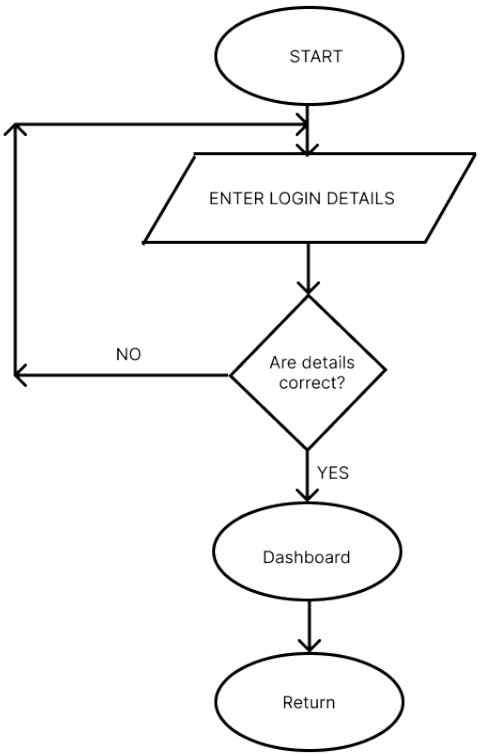


Figure 4: Login Page

In login page, student enters his/her login details and he/she is validated on the basis of username and password and is redirected to dashboard.

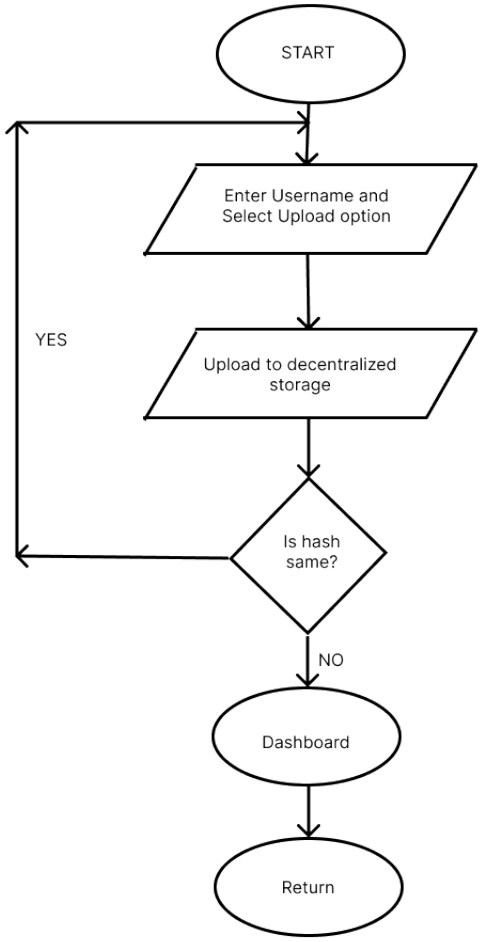


Figure 5: Upload Page

In upload page, admin enters username of the student and uploads the transcripts in the IPFS. After this, user gets the access of the transcripts after getting file uploaded.

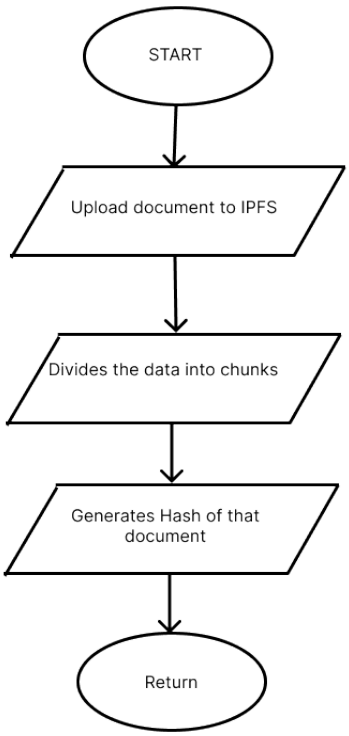


Figure 6: IPFS Design

Here after uploading file from the user interface, the file is uploaded on the IPFs i.e. Interplanetary File System and since it is decentralized storage so all peers those have access to the file can access the transcripts from IPFS.

3.3 User Interface Design

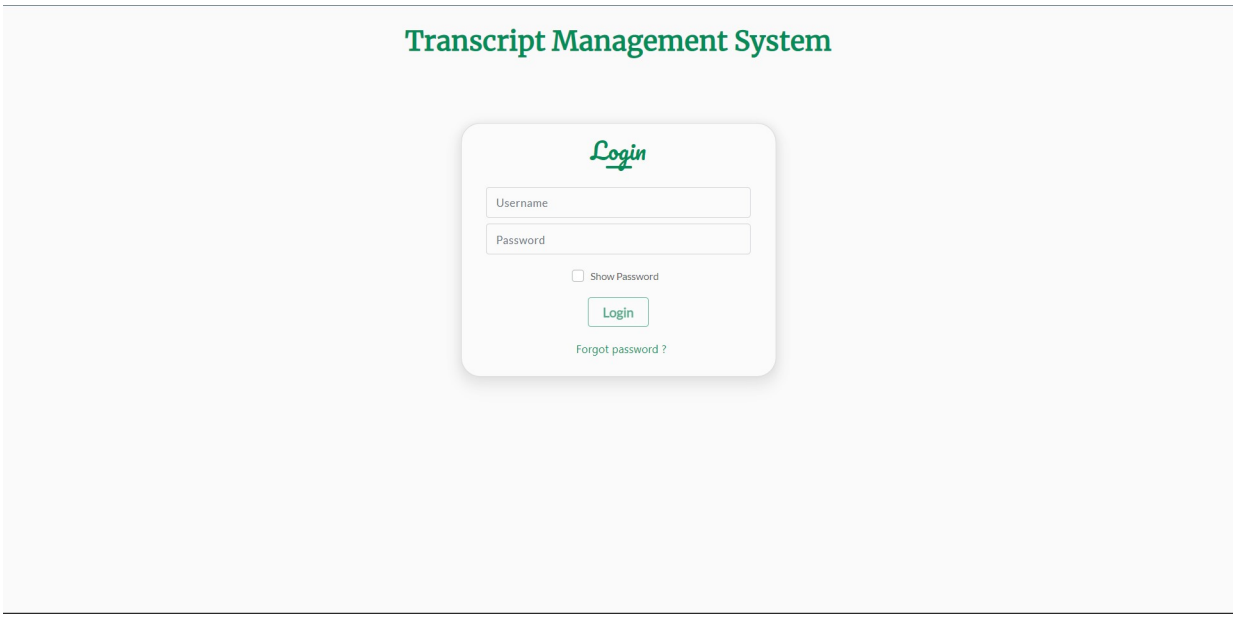


Figure 7: Login Design

In login page, student enters his/her login details and he/she is validated on the basis of username and password and is redirected to dashboard.

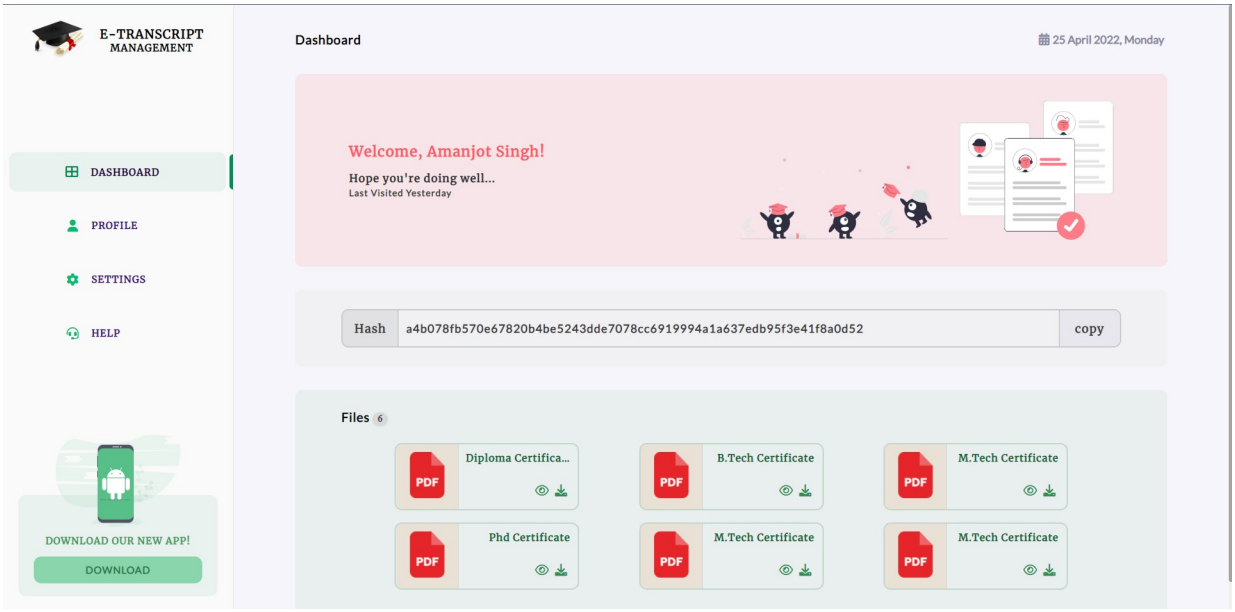


Figure 8: Dashboard Design

In dashboard, students can access their transcripts and can share the access to anyone they want without any delay. They can download and view direct from decentralized storage.

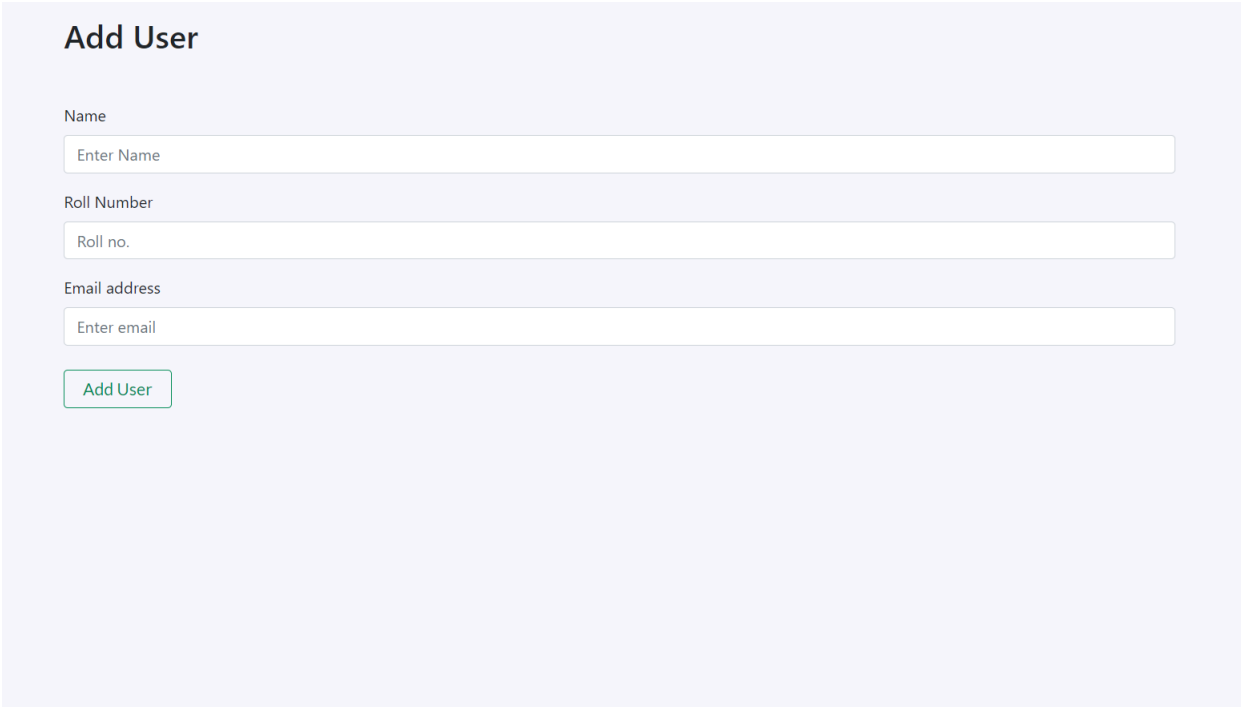


Figure 9: Add User Design

In add user page, user can be added to access the uploaded transcripts of that particular user only.

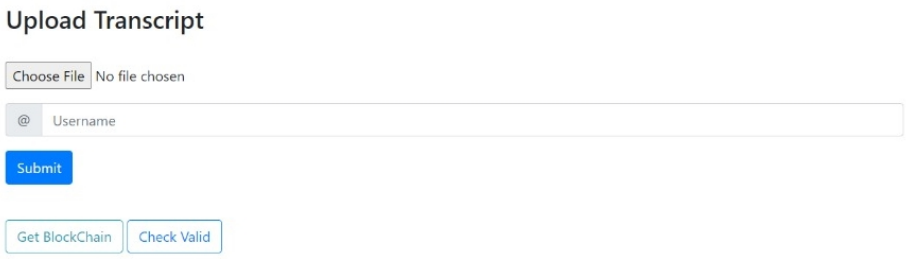


Figure 10: Upload Page Design

Upload page is designed for admins to upload the transcripts and show them to that particular user only.

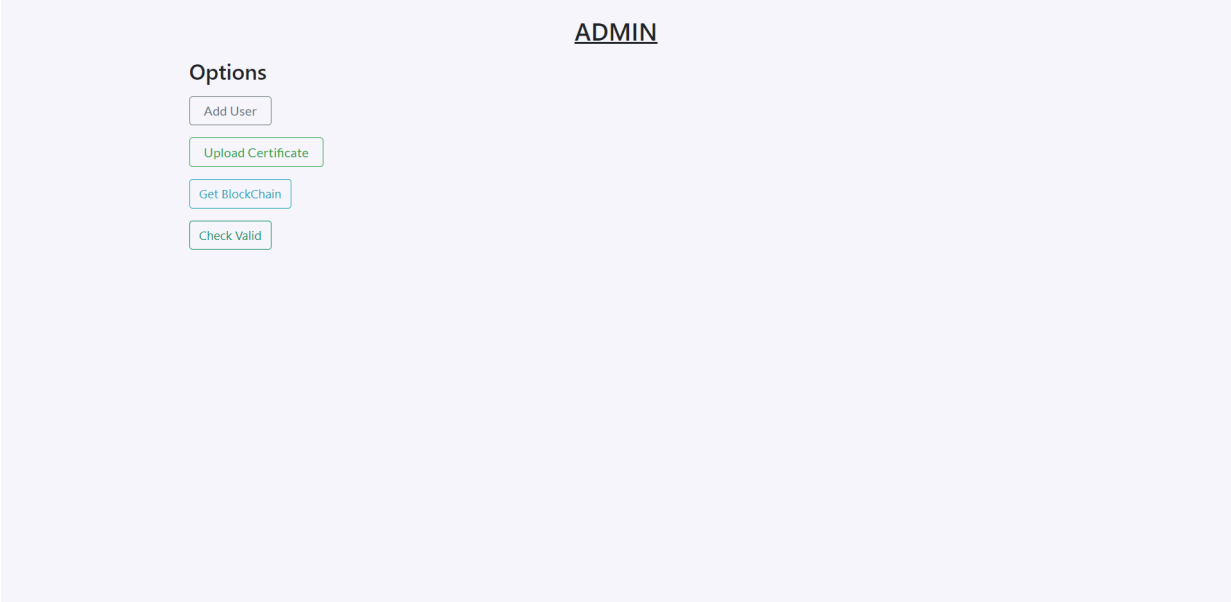


Figure 11: Admin Page Design

Admin Page is designed for admins to select the option between Get chain, Is chain valid, upload certificates, and add user.

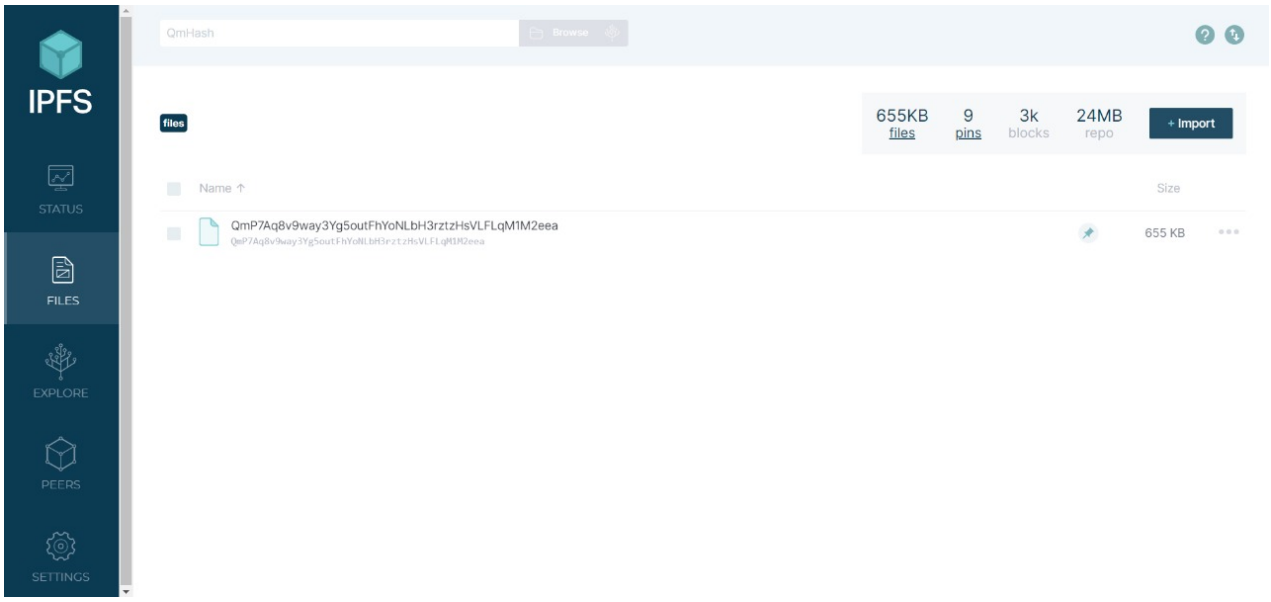


Figure 12: IPFS Design

3.4 Decentralized Storage Design

1. John wants to Upload file on IPFS.
2. He encrypted it with Jinny's public key.
3. He uploaded it onto the IPFS.
4. File is available on the IPFS network.
5. Jinny can access that file with her private key.
6. No one else can decrypt the file.

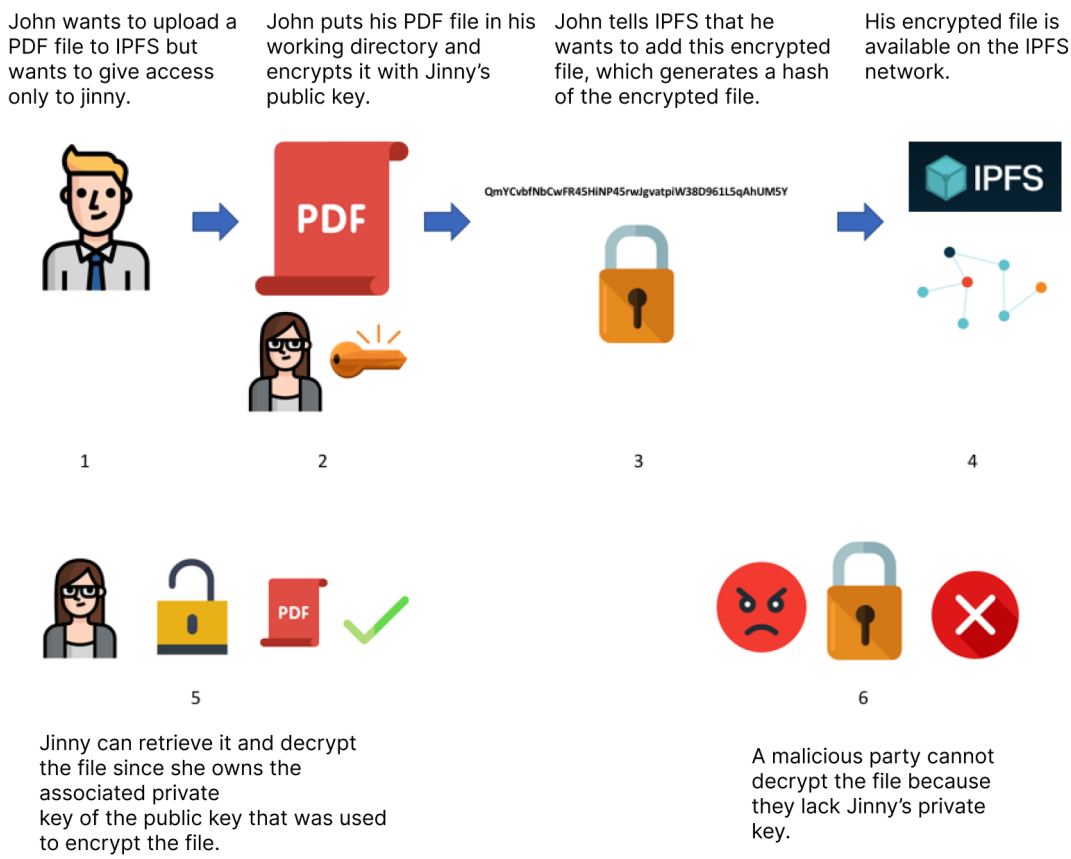


Figure 13: IPFS Workflow

3.5 Database Design

- Student Table of handling database of students with their certificates:

Table name: main WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL
1	name	TEXT					
2	email	TEXT					
3	username	TEXT					
4	password	TEXT					
5	encryKey	TEXT					
6	listOfBlocks	TEXT					
7	hash	TEXT					

Figure 14: Student Database

- Table of saving blockchain in the database:

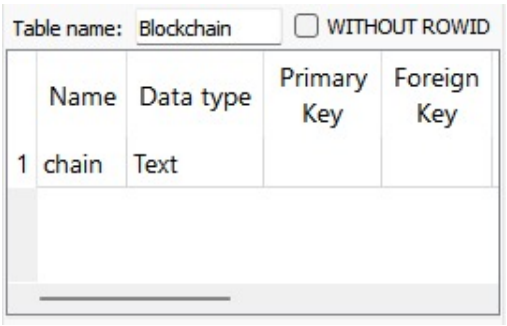


Figure 15: Blockchain Table

3.6 Methodology

We have chosen the **Waterfall model** as our **Systems Development Life Cycle** (SDLC) as it has been observed to best fit the nature of the research since it provides an easy to understand sequential and systematic process of developing the software that had the possibility of the changing requirements.

The waterfall model software development life cycle traditionally have the following five stages:

- **Data Collection of Transcripts:** To collect the transcript data to be stored in blocks and to observe the data through repeated careful listening.
- **Creation of Nodes:** Blocks of data are stored on nodes. To connect all the nodes on blockchain so they can constantly exchange the latest blockchain data with each other so that all nodes stay up to date.
- **Creation of Blocks:** To create blocks in the blockchain and identifying then with SHA256 values that will include encrypted transaction information from previous blocks and new transaction information.
- **Blockchain:** To build chain of blocks so that when a new transaction occurs, a block is added in the blockchain.

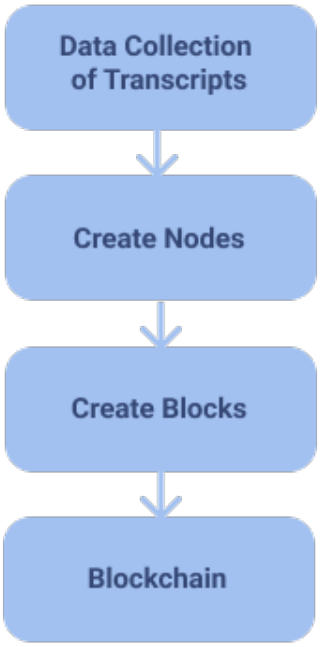


Figure 16: Methodology of work

4 Implementation, Testing, and Maintenance

Pilot implementation of the project has been done with many features and interactive interface. Then we have done Testing by adding the users, uploading the transcripts of the particular user and then viewing the transcripts after login.

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

- Python Language: Python is a widely used general-purpose, high level programming language. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

- IPFS: Interplanetary File System is a peer-to-peer protocol where each node stores a collection of hashed files. A client who wants to retrieve any of those files enjoys access to a nice abstraction layer where it simply needs to call the hash of the file it wants. IPFS then combs through the nodes and supplies the client with the file. It is a decentralized way of storing and referring to files but gives you more control and refers to files by hashes, allowing for much richer programmatic interactions.

- Flask: Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

- HTML: HTML stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages. It describes the structure of a Web page. It

consists of a series of elements. HTML elements tell the browser how to display the content.

- SASS: Sass stands for Syntactically Awesome Stylesheet. It is an extension to CSS. It is a CSS pre-processor. It is completely compatible with all versions of CSS. It reduces repetition of CSS and therefore saves time.
- Javascript: JavaScript is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for web-pages. It is well-known for the development of web pages, many non-browser environments also use it.

4.2 Coding standards of Language used

- Writing Well-Structured Code
- Having Proper Comments
- Proper Naming of Variables, Classes, Functions and Modules
- Writing Modular Code
- Exception Handling for every critical situation
- Use the 'with' statement while opening a file, the 'with' statement closes the file even if there is an exception raised

4.3 Project Scheduling using various tools such as PERT, GANTT charts, Open PROJ etc.

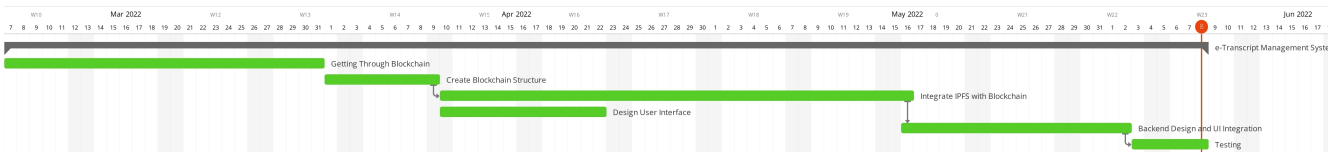
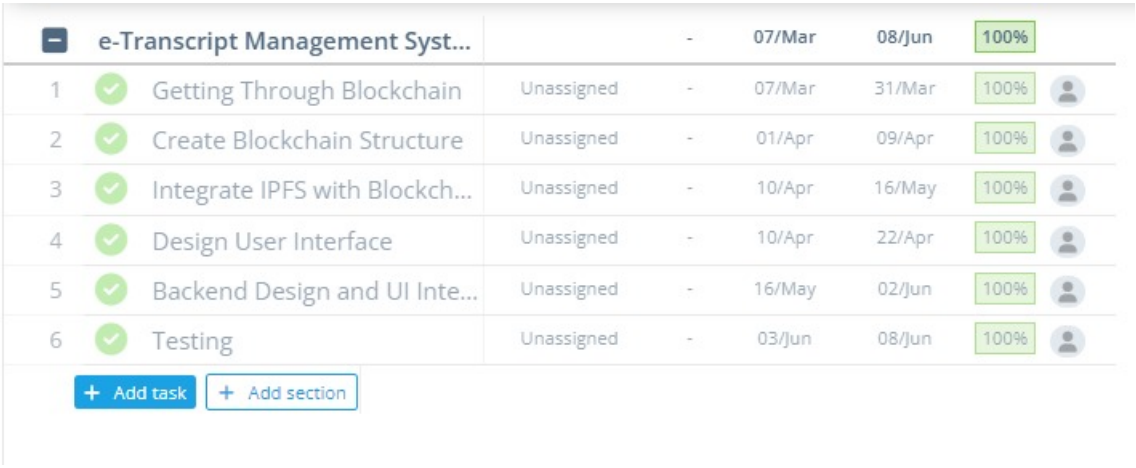


Figure 17: Gantt Chart



4.4 Testing Techniques and Test Plans

We have prepared these testing techniques and test plans:

- **Functional Testing:** Functional Testing plays an important role in Blockchain Testing as it helps in evaluating business requirements, processes, and effectiveness of use cases. Below are the components that can be tested as part of functional Testing:

  1. Blockchain has been tested by adding blocks and then also verifying whether our blockchain is valid or not.
  2. Also it has been tested that if we chain data of one block then whether it will affect the hash of that block or not.
  3. If we will add a block, it will add to the next of last block or not.
  4. Whether Each student can access his/her certificates or not from our website.

- **Integration Testing:** Blockchain application work in multiple environments. So, it is important to test inter-system connections. We have tested whether our project working fine after integrating every components or not. Whether it is encrypting the file while uploading on IPFS, after it got integrated with whole project.

- **Performance Testing:** It helps in identifying hardware and software bottlenecks in advance. It helps to figure out the potential costs of running the application in the cloud or other environments.

- **Node Testing:** All diverse nodes on the Network has been tested independently to ensure smooth cooperation.

- **API testing:** Application Programming Interface tests the interaction between applications in the blockchain ecosystem. API Testing ensures that requests and responses are formatted and operated properly. For this, we have tested whether our data that we are collecting from , whether they are submitted in the database or not using post request and dashboard is able to get the data when it will be opened or not.

5 Results and Discussions

5.1 User Interface Representation

We have made a website for showing the project working.

5.1.1 Brief Description of Various Modules of the system

Class of Blockchain having many functions:

- Create Blockchain: Returning a block which is the having all information about it.
- get previous block: It will return hash of previous block.
- hash: Return sha256 hash of encoded block.
- is_chain_valid: It will return True if the chain is valid else False.
- file_to_sha256: It will convert file into hash and return hash value of document.
- encrypt_file: This function will encrypt our file using private key of particular user and return encrypted file and key.
- decrypt_file: This function will decrypt the file using private key of particular user and write decrypted content int the file placed that we will pass through function.
- save_chain: This function will save chain in the database so that it will be accessible when we rerun our project.
- is_document_valid: It will return true if document is valid else false.

5.2 Snapshots of system with brief detail of each

The website includes many pages

- Dashboard page It includes showing certificates of particular user. There is also a option through user can preview his/her certificate and even he/she can download it. There is also hash of particular user from where one can copy.
- User can login using username and password and then after checking password from the back-end, if it's correct then he/she can be redirected to the dashboard page.
- Add User: Admin can add the student using student's name, CRN and email.
- Upload Certificate: Admin can upload username's certificate through block-chain in a securely manner.
- Admin Page: On this page, admin can select option between Add User, Upload Certificates, Get chain, Is Chain Valid.

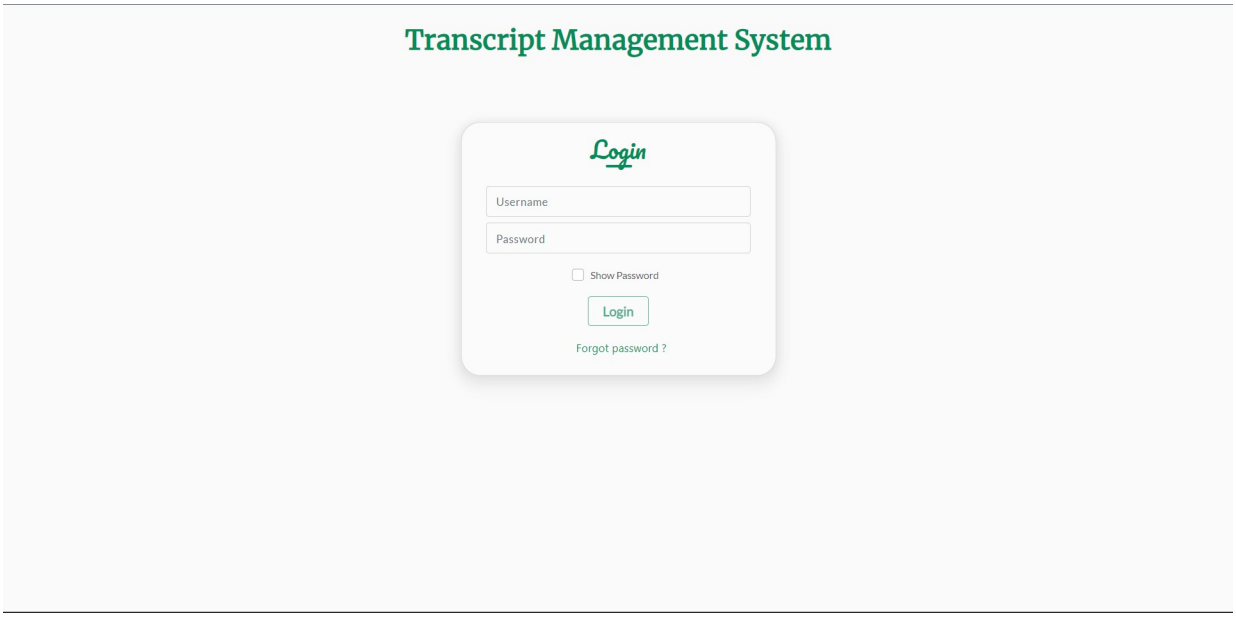


Figure 18: Login Page

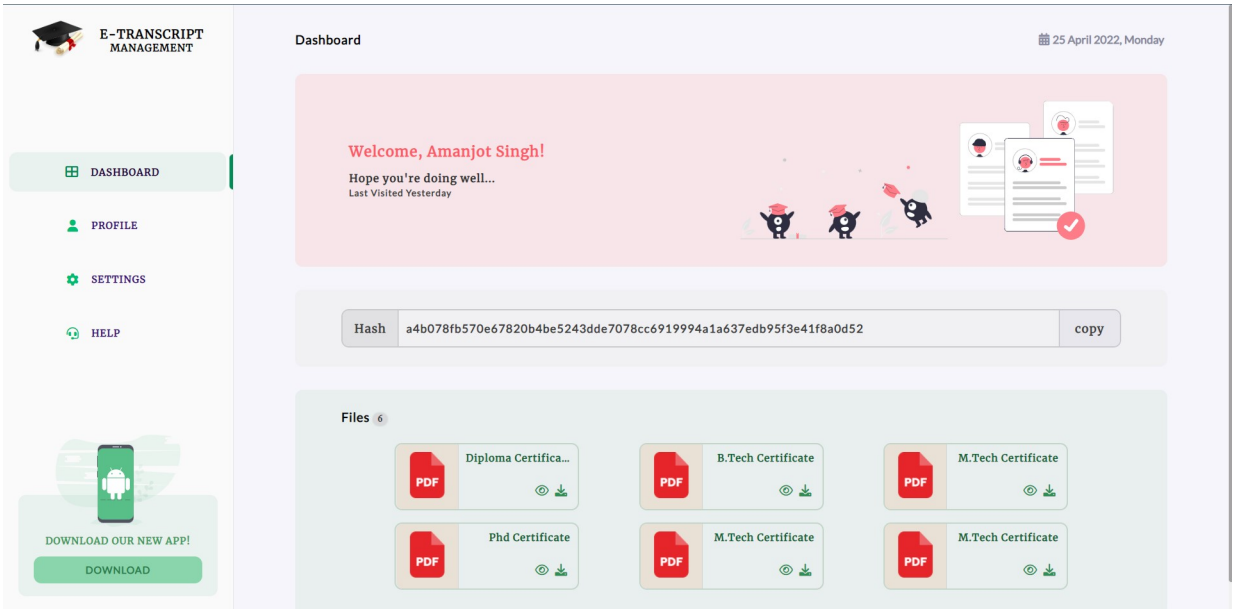


Figure 19: Dashboard Design

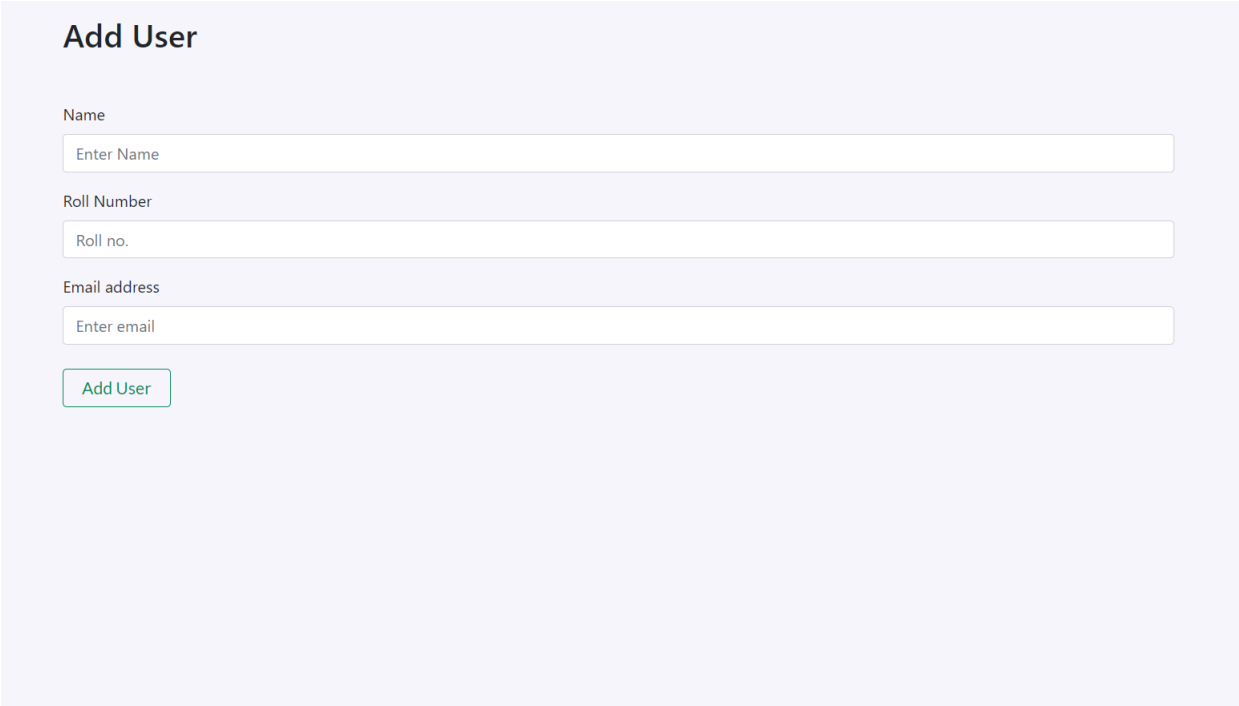


Figure 20: Add User Design

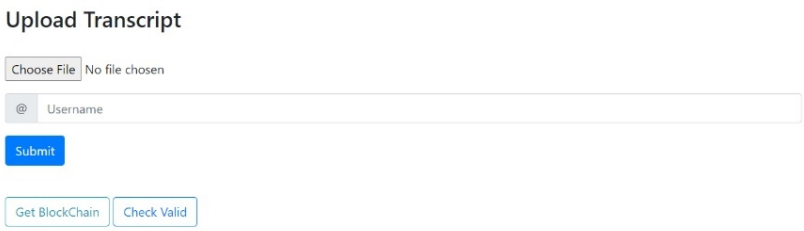


Figure 21: Upload Page Design

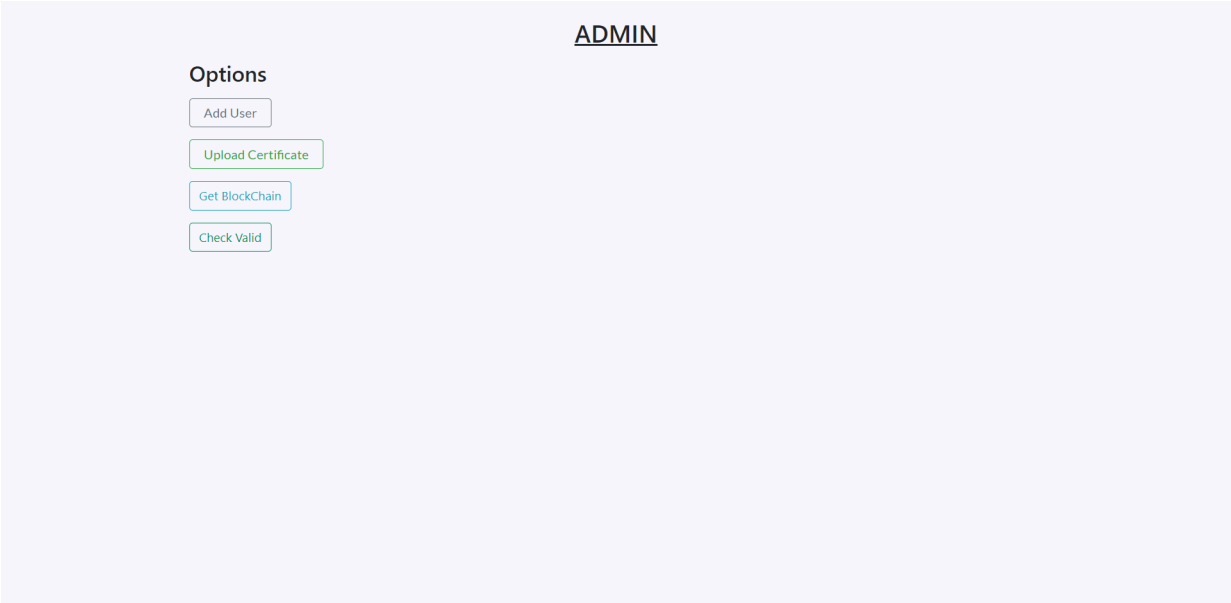


Figure 22: Admin Page

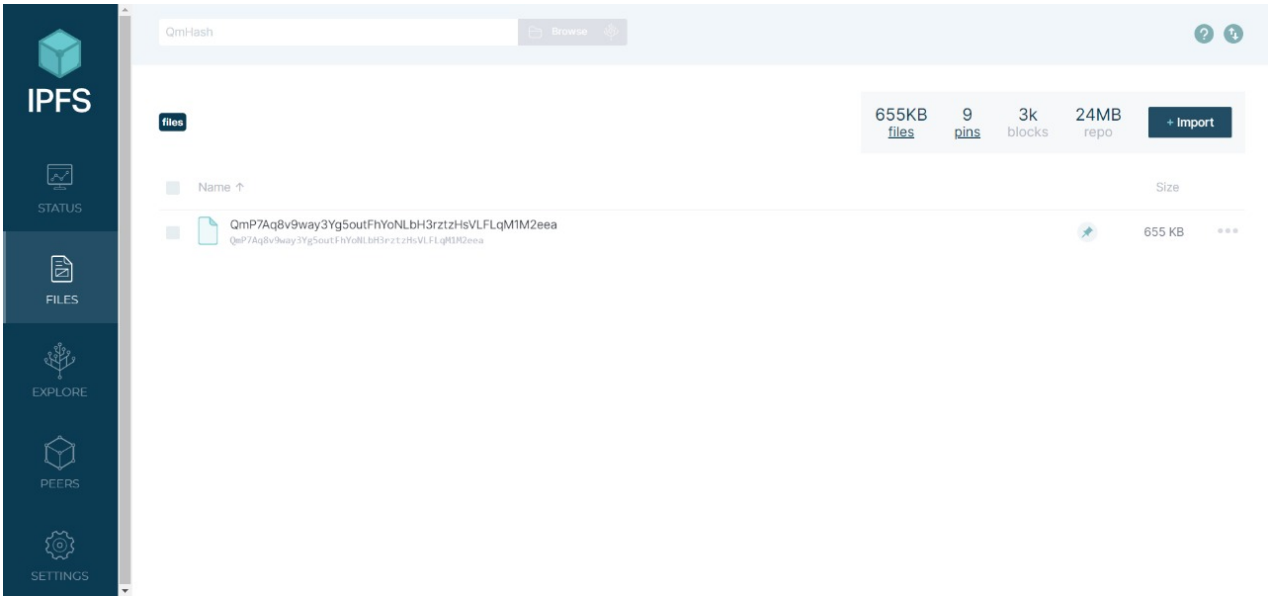


Figure 23: IPFS Design

5.3 Back Ends Representation (Database to be used)

We have used the SQL Database and Flask for our project. Because SQL Database is used for having database of students. Main Fields Included:

- Name
- CRN
- Email
- Password

- ListOfBlocks
- EncryptPrivateKey

5.3.1 Snapshots of Database Tables with brief description

- Name of the Student is of datatype text
- Class Roll No. of the student is of datatype text and it is a primary key.
- Email of the student is of datatype text
- Password of the student through which a student can login to view his/her certificates. It is of datatype text.
- ListOfBlocks: Indexes of the blocks in the block-chain having the hashes of the files uploaded on IPFS. It is of datatype text.w
- EncryptPrivateKey: Private Key which can be used for encrypting the file and decrypting the file that has been uploaded on IPFS.

Table 1: Blockchain Database

Chain
1, 2
3, 4

Table 2: Student Database

Name	Email	Username	Password	Encryption Key	List of Blocks	Hash
Vrishti Gupta	ivrishtigupta@gmail.com	1915086	vrishti123	$07eVh2QgGdZeMqOM8FQCw$	2,5	6d8ace04726c183856a 0fcbc17b65cc f5bbfbad5afa1d3 3a2e0acda487267281
Priyanka Jhamb	priyankajhamb73@gmail.com	1915103	priyanka123	$07eVh2QgGdZeMqOM8FQCw$	3,6	6d8ace04726c183856 a0fcbc17b65cc f5bbfbad5afa1d3 3a2e0acda487267281
Amanjot Singh	amanjots726@gmail.com	1915105	amanjot123	$07eVh2QgGdZeMqOM8FQCw$	4,7	6d8ace04726c183856 a0fcbc17b65cc f5bbfbad5afa1d3 3a2e0acda487267281

6 Conclusion and Future Scope

We have made a pilot implementation of the project E-transcript Management using Block-chain. In this, we are storing the transcripts on the block chain and each student can access only his/her certificates using amazing interface of website. Admin can upload the certificates in the block-chain using website page and maintain database of blocks in front of the username. If any university want to verify the certificate, student can send his/her personal hash with the university and verification can be done. Admin can add users through web page interface. Also each student need to login the website to check his/her certificates.

In future, we can build up for the college so that College can store the certificates securely on the block-chain and no one can alter any certificate because of the feature of block-chain. If anyone will do it, then it can be detected. Verification of the certificates process will be fast in the future after using this project.

7 References

[1] Timothy Arndt and Angela Guercio. Blockchain-based transcripts for mobile higher-education. International Journal of Information and Education Technology, 10:84–89, 01 2020

[2] Sura I. Mohammed Ali, Haitham Farouk, and Hussien Sharaf. A blockchain-based models for student information systems. Egyptian Informatics Journal, 2021

[3] Antonio Bartolomé, Carles Torlà, Linda Castañeda, and Jordi Adell. Blockchain in education: Introduction and critical review of the state of the art. 61, 01 2017.

[4] Benjamin W Cashman Richard S. Clark Ronald P. Uhlig, Rich Yonts and Brett Niema, Blockchain in Education: Introduction and Critical Review of the State of the Art, 2017

[5] Blockchain Course: https://www.udemy.com/course/build-your-blockchain-az/

[6] Proof of authority Online Material:

- Wikipedia,
- https://academy.binance.com/en/articles/proof-of-authority-explained ,

[7] IPFS Documentation: https://ipfs.io/

[8] https://medium.com/python-pandemonium/getting-started-with-python-and-ipfs-94d14fdf