A Project Report on

Online Certificate Generation & Verification using Blockchain Framework

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

Bachelor of Engineering

in

Information Technology

by

Prasad Jadhav(17104003)
Rutwik Gaikwad(17104074)
Aseem Godambe(17104058)

Under the Guidance of

Prof. Kiran Deshpande

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2020-2021

Approval Sheet

This Project Report entitled ***"Online Certificate Generation & Verification using Blockchain Framework"*** Submitted by ***"Prasad Jadhav"(17104003),"Rutwik Gaikwad"(17104074),"Aseem Godambe"(17104058)*** is approved for the fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Kiran Deshpande
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Online Certificate Generation & Verification using Blockchain Framework"*** submitted by ***"Prasad Jadhav" (17104003), "Rutwik Gaikwad" (17104074), "Aseem Godambe" (17104058)*** for the fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2020-2021.

Prof. Kiran Deshpande
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

_____

(Prasad Jadhav 17104003)
(Rutwik Gaikwad 17104074)
(Aseem Godambe 17104058)

Date:

Abstract

In the modern digital age, the advancement in technology has resulted in solving complex problems. But this advancement has also resulted in increased illegal activities. One such major activity is forgery of documents and certificates of an individual. There are many cases reported of certificate forgery everyday and many of them go undetected. The purpose of this study is to develop a system that would authenticate certificates to their real owners and could verify the same on a specific portal. To achieve this, in this project we focus on Ethereum Blockchain as it is immutable, transparent, scalable, and also cost-effective. Universities would be able to generate their own certificates on the web portal itself by uploading their certificate template in Scalable Vector Graphics(SVG) format and the student data in Comma-separated Values(CSV) format. Each generated certificate would be allotted to the respective student within the system. Each certificate would be given a unique hash code so that it could be verified easily and there would be no scope of duplication. The authentication and verification would be done on the same web portal. To sum up, this method would save time and efforts that are required to verify a certificate manually and would result in an effective, secure way to generate certificates.

Contents

List of Figures

List of Tables

List of Abbreviations

SVG:	Support Vector Graphics
CSV:	Comma-separated values
QR code:	Quick Response code
OTP:	One Time Password
IPFS:	Interplanetary File System
HTTP:	Hypertext Transfer Protocol
URL:	Uniform Resource Locator
EVM:	Ethereum Virtual Machine

Chapter 1

Introduction

Everyone has a particular talent or excels in a field and certificates are a great way to reflect the achievements. It proves that an individual has acquired knowledge of a particular skill. Companies need skilled employees and certificates prove to be a valid solution to depict once capabilities. Displaying a certificate of a particular skill verifies that the certificate holder is competent in that skill and helps the company in the hiring process. Also due to the current pandemic situation, online learning and certifications through online courses has drastically increased. People have started to adapt to online learning and hence the certificates hold a greater value to validate the progress of an individual and also it adds a great value to their profiles.

As certificates hold such higher values people tend to misuse it and generate a fake copy of the certificate of the skill which they have not acquired. The companies or the organisation where an individual is applying offers an advantage to the person having the necessary certificates. It's difficult for the organisation to check individual certificates as they may have to contact the issuer and it takes a lot of time and effort. The fake certificates can be easily generated using any online website or normal photo editing software. As a result it makes it more difficult for the deserving candidate to have all the genuine certificates.A survey conducted by UK's national qualifications agency UK NARIC conducted across 17 countries in different universities found the common problem of difficulty in verifying documents. According to the survey, it has been found that 62% of them verify the documents by contacting the institutes awarding them while 14% of them didn't even bother to check the originality of the certificate. It stated that around 75% of the certificates submitted were fraud [1]. Similar research and investigation conducted by the BBC Radio 4's File on Four programme found out about thousands of fake degree certificates in the UK from "Diploma Mill" in Pakistan. This deteriorates the standard of employees being hired and unfair against the students who are hardworking and achieving the results [2].

The motive of this research is to create a system which is secure, easy to access and verify certificates. The organisation hiring an individual could verify the certificate online without the need to contact the issuer. Blockchain technology along with different hashing methods is used to maintain a secure record of the information. An SVG(Scalable Vector Graphics) format certificate template is used specifically to reduce the cost of storing multiple images on the blockchain network. This creates a scalable and economical system for an organisation to implement.

1.1 Scope

Manually generating and verifying a certificate takes a huge amount of time for any institution or an University. So due to this issue, having a single portal for certificate generation and verification will save a lot of time of both the Institution/University as well as the student.

1.2 Objectives

1. **Central Portal for all**: This system makes it possible for all the Universities/ Institutions to generate their certificates on one single web portal with their own private educational email Ids.

2. **Throughout data security**: Ethereum blockchain makes this system highly secure as the data cannot be overwritten and the certificate cannot be generated without the admin's permission.This system would also take care of a certificate's security as each certificate generated has a unique hash code making sure that there is no scope of duplicacy.

3. **Certificate generation**: Universities would be able to generate their own certificates on the web portal itself. Universities/Institutions will have to upload their certificate template in Scalable Vector Graphics(SVG) format and the student data in Comma-separated Values(CSV) format. Each generated certificate would be allotted to the respective student within the system.

4. **Easy to use**: It is ensured that the web portal is easily accessible and very easy to use. The web portal would have a very simple-to-understand user interface making sure that anyone could easily use it without any problem.

5. **Cost effective**: Ethereum Blockchain makes this system highly scalable. But Ethereum cost is a major drawback. To overcome this, the system would not upload the entire certificate on the blockchain and upload only the mapping data(data that is to be mapped on the certificate), since this mapping data is very less the cost of the Ethereum would be very low and affordable, thus making it cost effective.

Chapter 2

Literature Review

2.0.1 Shanmuga Priya R,Swetha N 'Online Certificate Validation Using Blockchain.' [6]

This document summarizes the problems of forged certificates and how blockchain can solve the issue. Netbeans IDE and Android Studio are used to develop the server communication and the application to scan the QR codes. EthereumJS is used for faster Ethereum applications. The application involves the user uploading his/her certificate like 10th-grade mark sheet, college certificates, government certificates, and so on to the portal. After uploading the certificate, the data is then sent to the issuer for validation. The issuer (example: School which is responsible for validation of 10th grade Marksheet) has to validate the data received by the application for verification. Once verification is successful, the data will be stored on the server else it would be discarded. On the mobile application, a QR code will be generated based on the certificate number. The QR code can then be shared with anyone else for verification in case of necessity. When the QR code is scanned, an OTP (One Time Password) will be sent to the registered mobile number for verification. After proper authentication, the user can view the certificate. If the number of scans goes beyond the permitted limit, the location of the scanner will be sent to the authorized user with a permission link. From that link, the authorized user can either allow or deny the person. The major drawback of this system is that verification can be delayed by the issuer as the certificate has to reach the organization for verification and the data is sent to blockchain only after verification. The OTP system can get tedious. This system can get costly as well.

2.0.2 Nitin Kumavat, Swapnil Mengade, Dishant Desai, Jesal-Varolia 'Certificate Verification System using Blockchain' [7]

This document summarizes that during the course of education the students achieve many certificates. Students produce these certificates while applying for jobs in public or private sectors, where all these certificates are needed to be verified manually. There can be incidents where students may produce fake certificates and it is difficult to identify them. The solution proposed in this system uses Ethereum and IPFS (Interplanetary File System). IPFS is a peer to peer, content address system. It is very similar to BitTorrent and MerkleDag. Unlike HTTP which restricts or provides low latency on transfer of large amounts over the network which uses IP addressing, IPFS uses content addressing. As a result it creates a distributed system of different nodes across the network. It returns us a hash value and it uses this value to retrieve the data. They propose to add the data with the certificate on the IPFS network to generate the hash for it and later on add the data to the blockchain network with the help of EVM. This topology includes two distributed networks to provide additional security to the model. While retrieving the certificate they would compare the hash and pull the entire certificate which has been stored on the IPFS network. The advantage of this system was it is more secure, provides reliability. With two distributed networks, it's almost impossible to tamper the data. On the other hand, the disadvantage of the system is that IPFS is a tedious process to set-up as well as storing the certificate image on any distributed blockchain platform costs a lot of money and hinders the scalability of the system.

Chapter 3

Proposed System Architecture

3.0.1 Methodology

This proposed system revolves around creating a secure and fast method to generate and verify the certificates using the features of SVG and blockchain. The certificate's data is hashed and stored on a blockchain network to provide security and immutability to the data. On the blockchain network, this data is stored on blocks. The block contains the hashed data, timestamp, and the id of the next block. Then the block is added to the network. The template of the certificate which is in SVG format is stored on a database server that is not a blockchain-based network as the certificate template won't be modified. The data which is hashed includes the name of the certificate holder, their email, issuing date, expiry date for the certificate, and information regarding other fields of the certificate. This data can only be submitted by the certificate issuer and thus ensuring the right data is provided. Anyone can check certificates of the students using the unique id provided to each student for their certificates or from a student's profile.

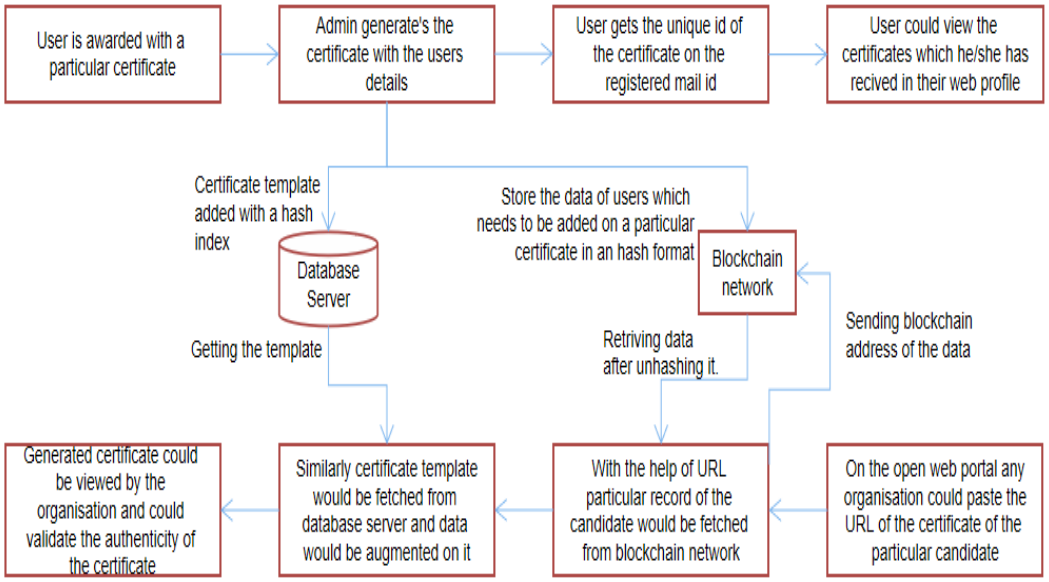


Figure 3.1: Architecture of proposed system.

3.0.2 Certificate Generation

Certificates can be generated only by the organizations registered on the website. They need to upload the template of the certificate in the SVG format. The SVG template should have a proper id for a particular field like the space for the name of the receiver should have a well-defined id, for example, 's_name'. Once the SVG template is uploaded the organization needs to upload a CSV file that has the record of the students who have won the certificates. The CSV needs to have the first row as the ids of the fields marked on the SVG template, along with it the CSV needs to have a compulsory email column that would contain the email of the receiver. Other columns of the CSV would be the remaining values that needed to be filled on the template like the receiver's name, issuing date and expiry date of the certificate, etc. The only condition is that the id of the field written on the SVG template should be the column header of the CSV file. This method is done to ensure that the blanks on the template could be filled by the right data in the CSV file. Once the upload action of the template and CSV file is done the certificates would be auto-generated for all the students receiving the certificates and would be available in their profile on the website. This would be done with the help of email ids associated with each student present in the CSV file.

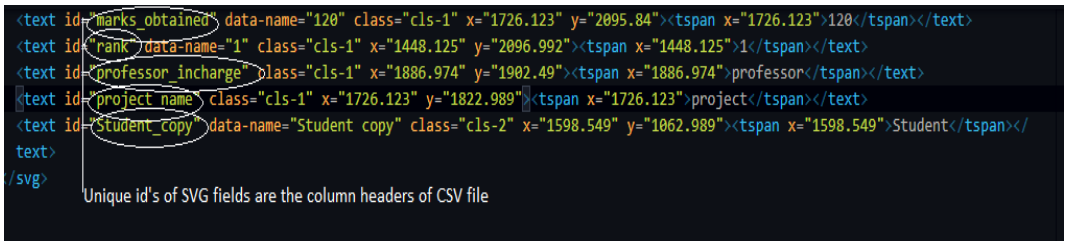


Figure 3.2: Example of a certificate template in an SVG format indicating the ID's of the fields of the certificate.

cert_id	email	Student_copy	marks_obtained	rank	professor_incharge	project_name
1	abc1@gmail.com	Rahul Sharma	70	2	Prof. Gopal Yadav	Water sanitisation
2	abc2@gmail.com	Vishal Pande	79	1	Prof. Gopal Yadav	Soil quality analysis

Figure 3.3: CVS file with the data of students winning the certificate for the above given SVG template.

3.0.3 Certificate Validation

In this process, the certificates are validated. As the data is stored on the blockchain network there is no way to edit the data once it's appended into the blocks on the network. The process works as when the certificate data is uploaded by the organization in CSV format the data is extracted from the CSV and uploaded on the blockchain network. Once the data is uploaded on the network, students who have won those certificates are informed through an email that they can view their certificates now in their profile and are provided a unique key or URL for the certificate. Students could use or provide this key, URL to anyone to authenticate the certificates. When a particular key is used the data of that candidate is retrieved from the blockchain and the corresponding template is fetched from the database

server. The data from the blockchain is mapped on the SVG templates as the fields of the template are pre-recorded and thus generates a certificate in front of the user which is not tampered with by any means. Thus an entity could verify that the certificate is genuine and possessed by the actual owner of the certificate.

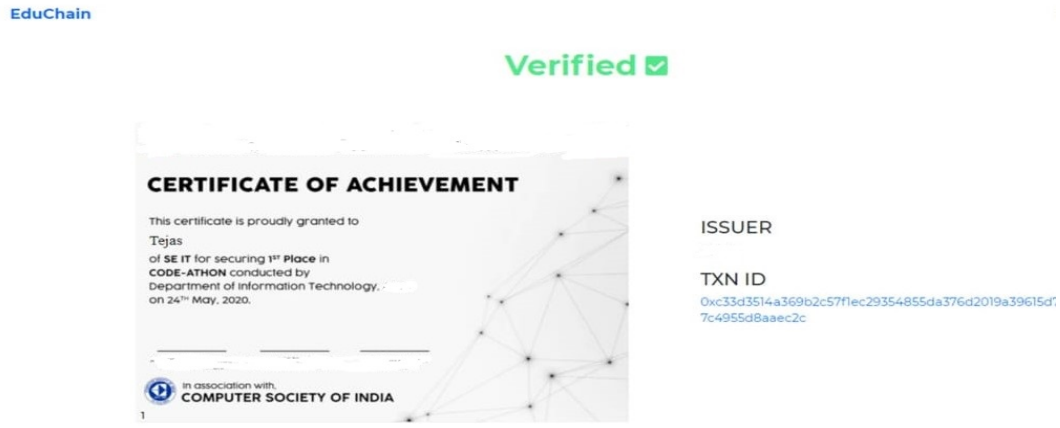


Figure 3.4: Certificate uploaded on the web portal which is verified and genuine certificate.

3.0.4 Working of Application

In our application on the landing page any individual could see a certificate by just entering its unique id on the search bar. There is an admin login, once logged in the admin one could add certificate and CSV file to generate certificates. Once the data is uploaded by the admin it would be stored temporarily in a database and would be uploaded on the blockchain network periodically using a CRON script every day at midnight. This is done to ensure every certificate's data is uploaded on the blockchain network because if we fail to upload any data we cannot rewrite the blocks on the network. Once the data is uploaded students are informed that their certificate is uploaded. When logged in as students they could see the certificates achieved by them in their profile and would see a green tick indicating that the certificate is validated. Students can share their certificate links with anyone to view their certificates.

Figure 3.5: Homepage of the web portal where anyone could search for any certificate to verify it's authenticity.

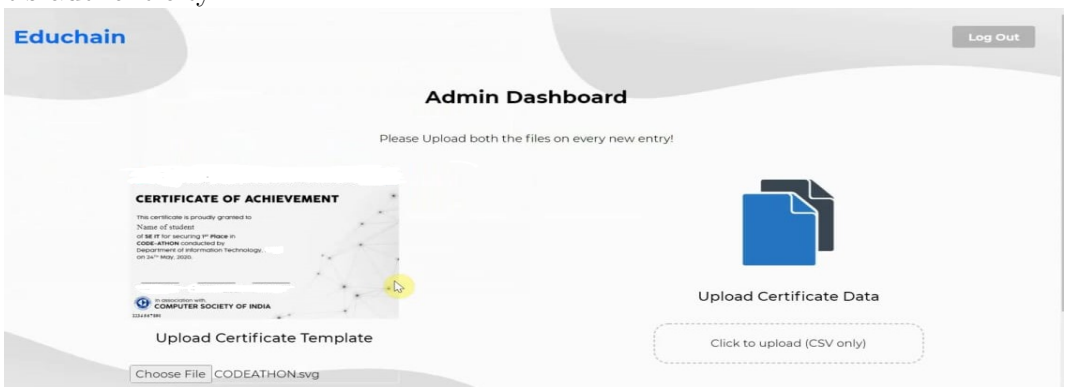


Figure 3.6: Admin dashboard to upload SVG and CSV file

Chapter 4

Design

4.0.1 Use-case Diagram

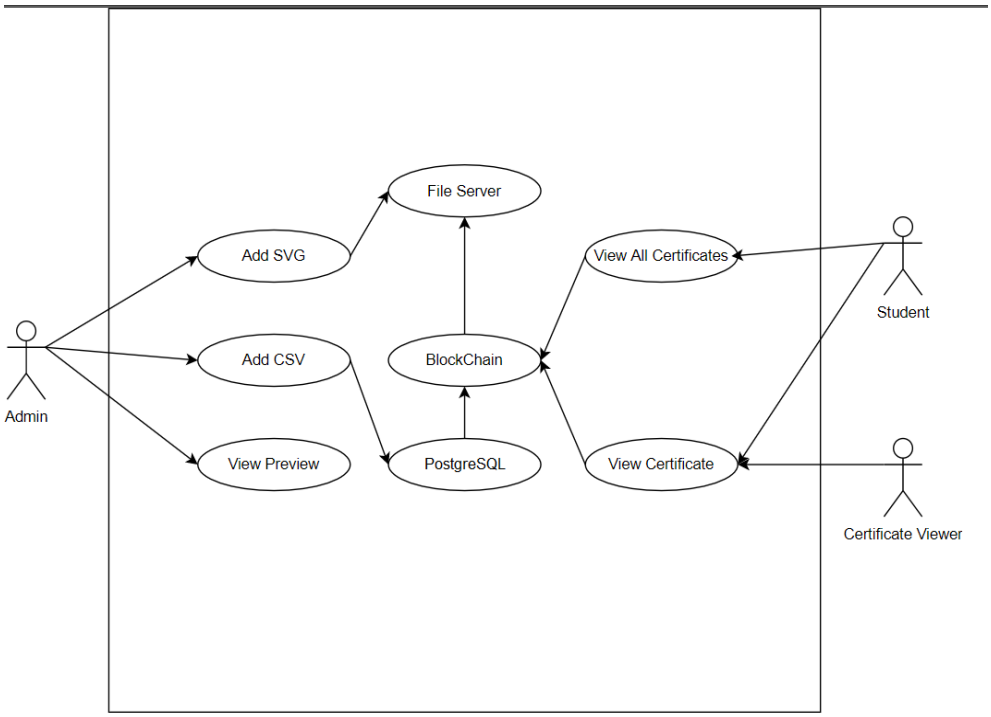


Figure 4.1: Use-case of the system.

In the above figure admin, students and certificate viewers are actors. In case of admin he/she can add SVG template for the certificate, add CSV file and can view the preview of the certificates. The students can interact with the system by viewing the certificates. Anyone else who has the certificate id can also view any particular certificate who acts as a viewer.

4.0.2 Database Schema Diagram

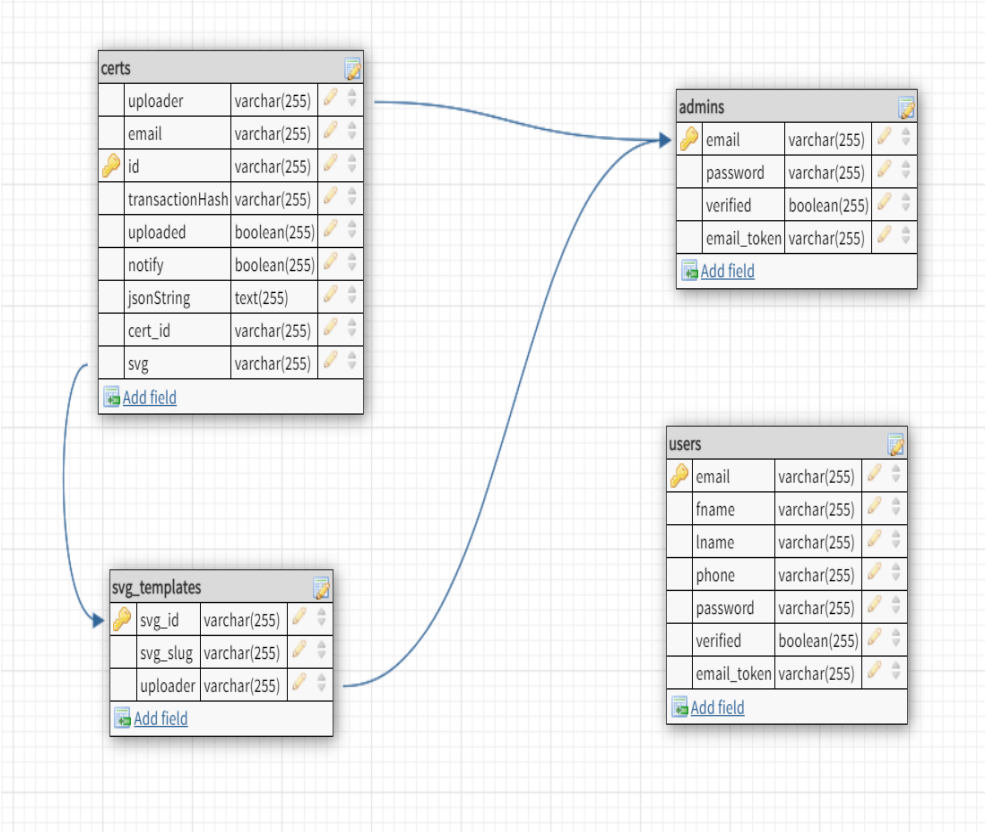


Figure 4.2: Database Schema diagram of the system.

The above figure depicts the database schema of the system and the relation between different entities in the system.

Chapter 5

Implementation

Figure 5.1: Program for admin to add SVG template file (a).

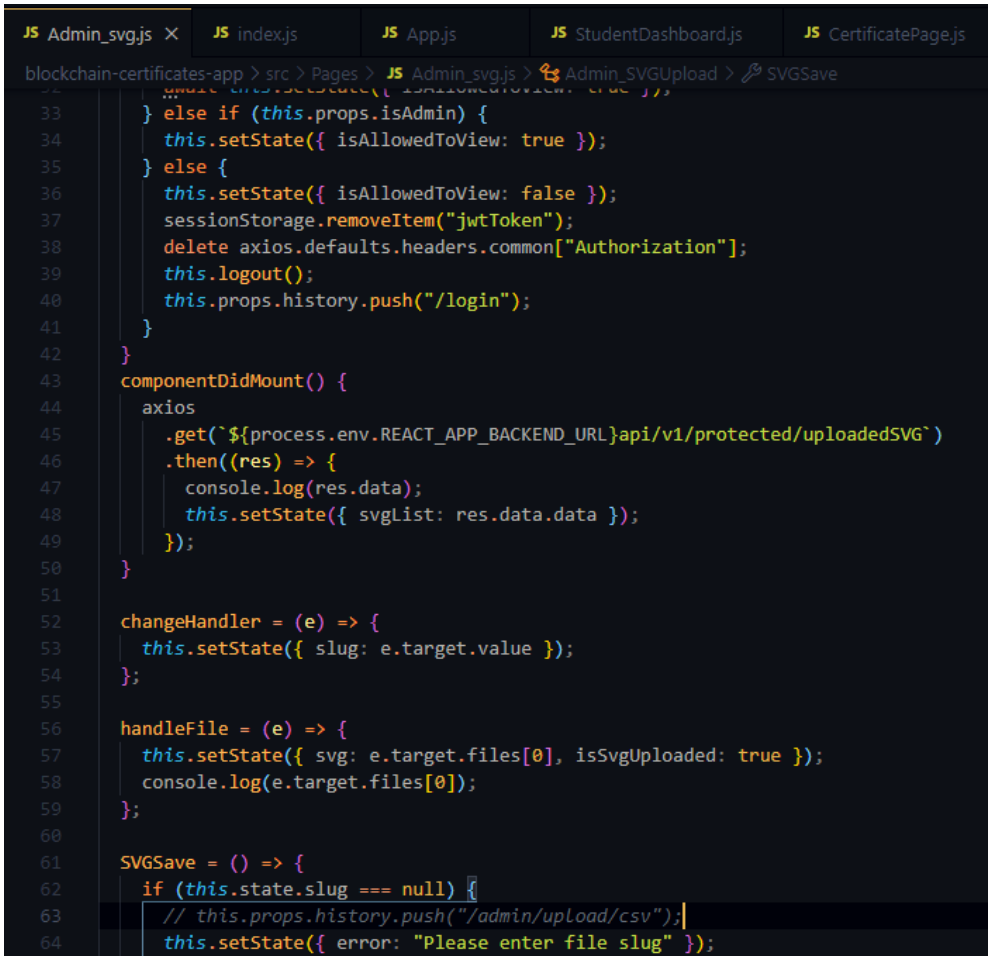


Figure 5.2: Program for admin to add SVG template file (b).

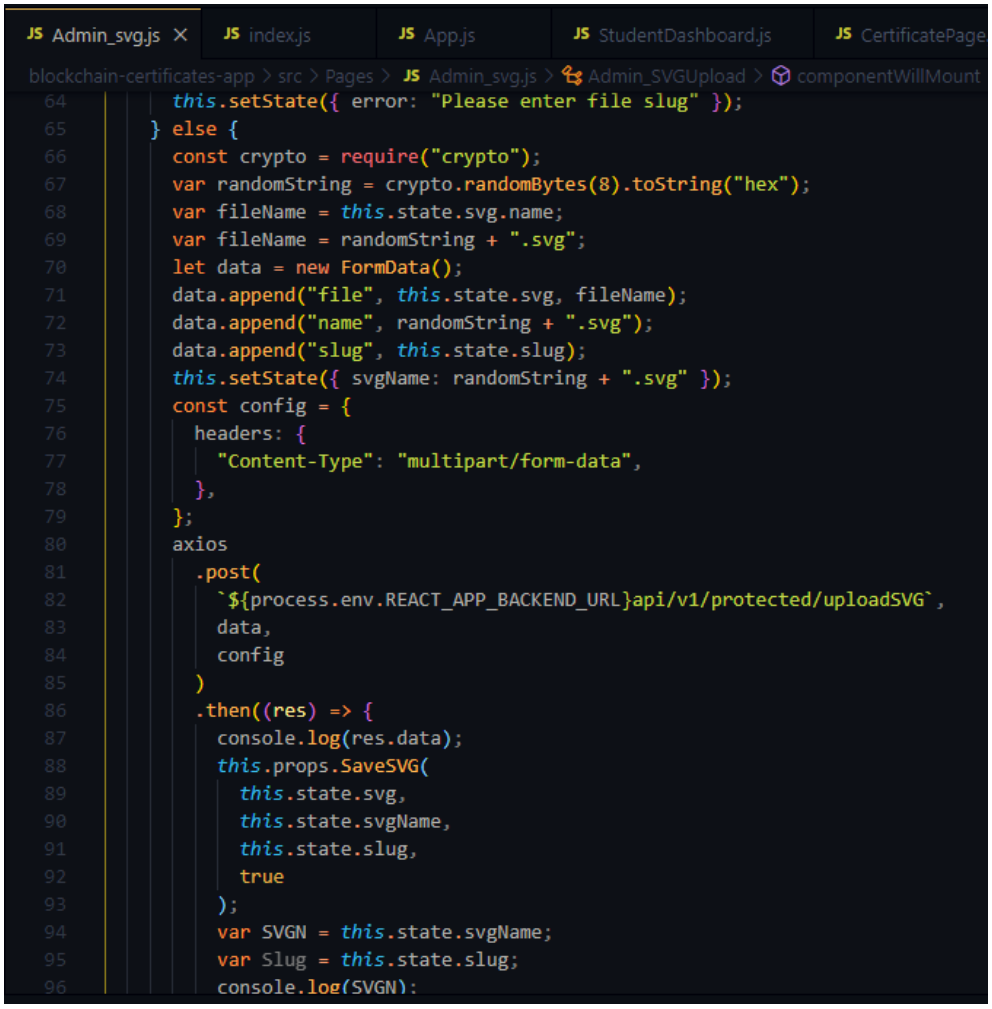


Figure 5.3: Program for admin to add SVG template file (c).

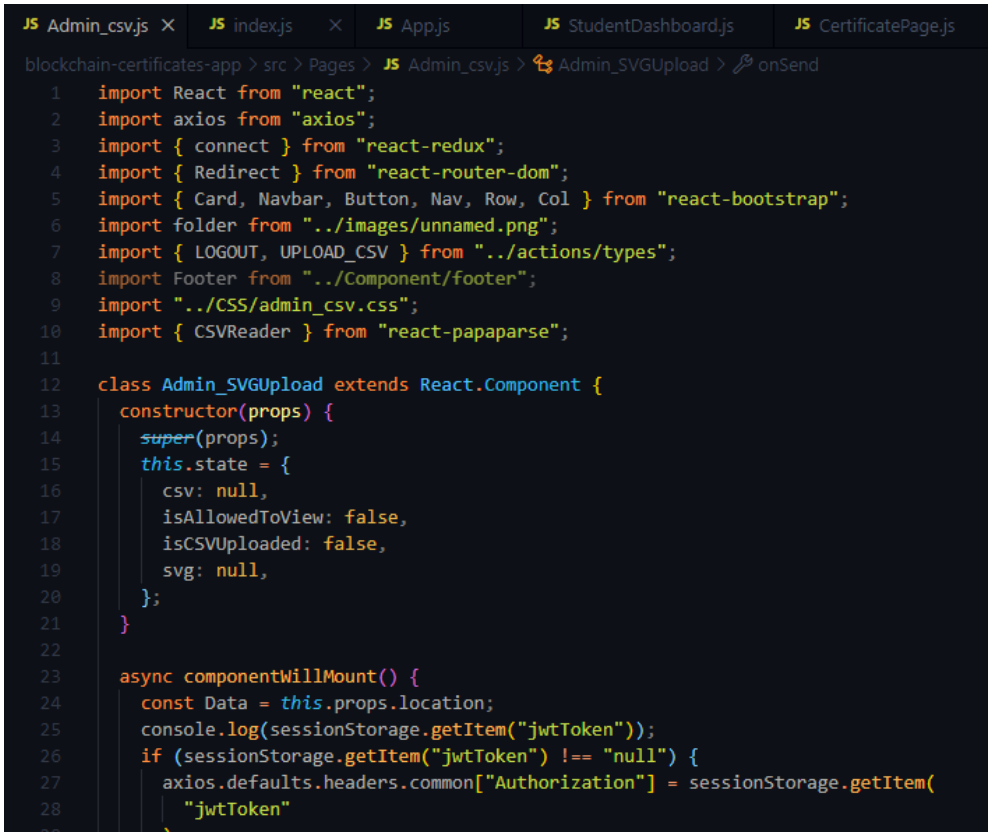


Figure 5.4: Program for admin to add CSV file (a).

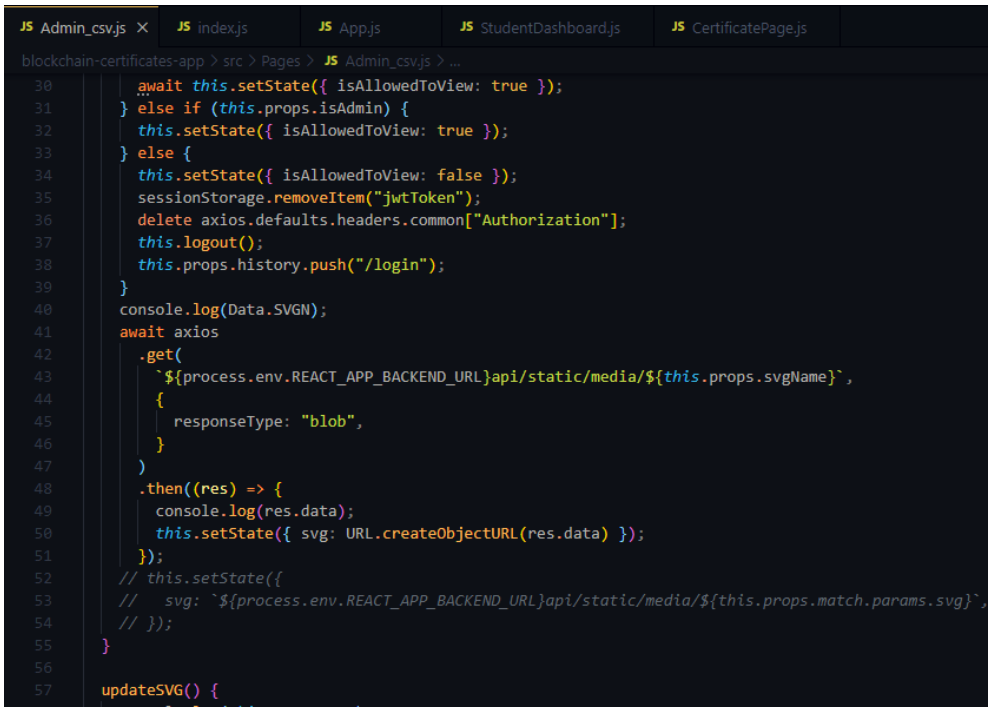


Figure 5.5: Program for admin to add CSV file (b).

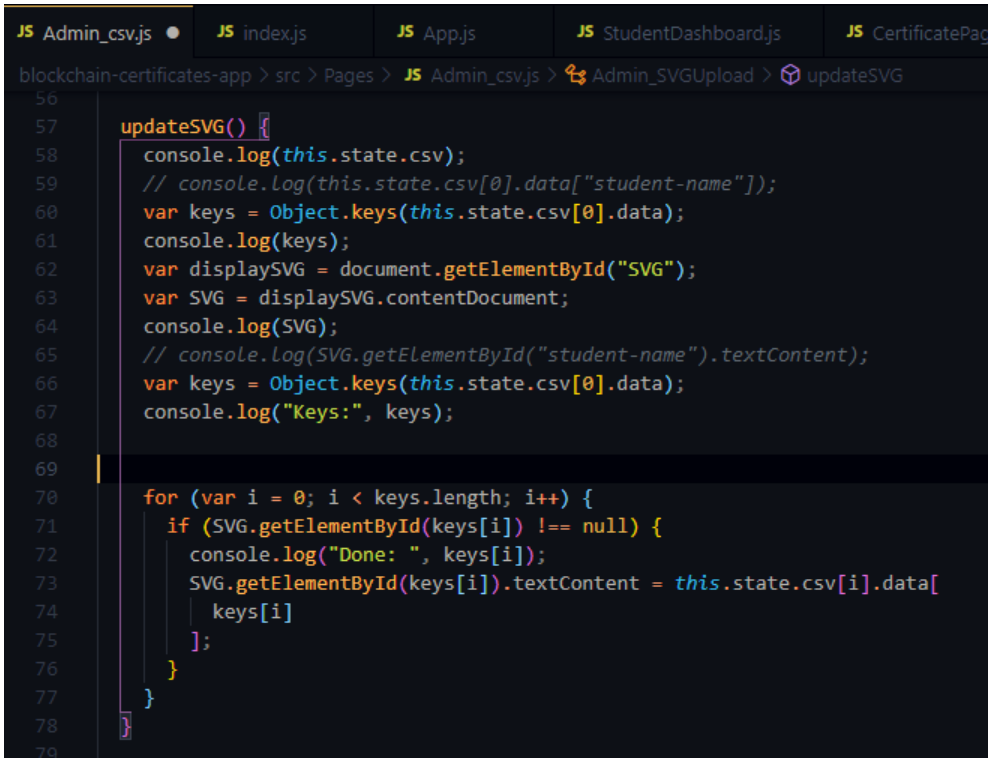


Figure 5.6: Program for admin to add CSV file (c).

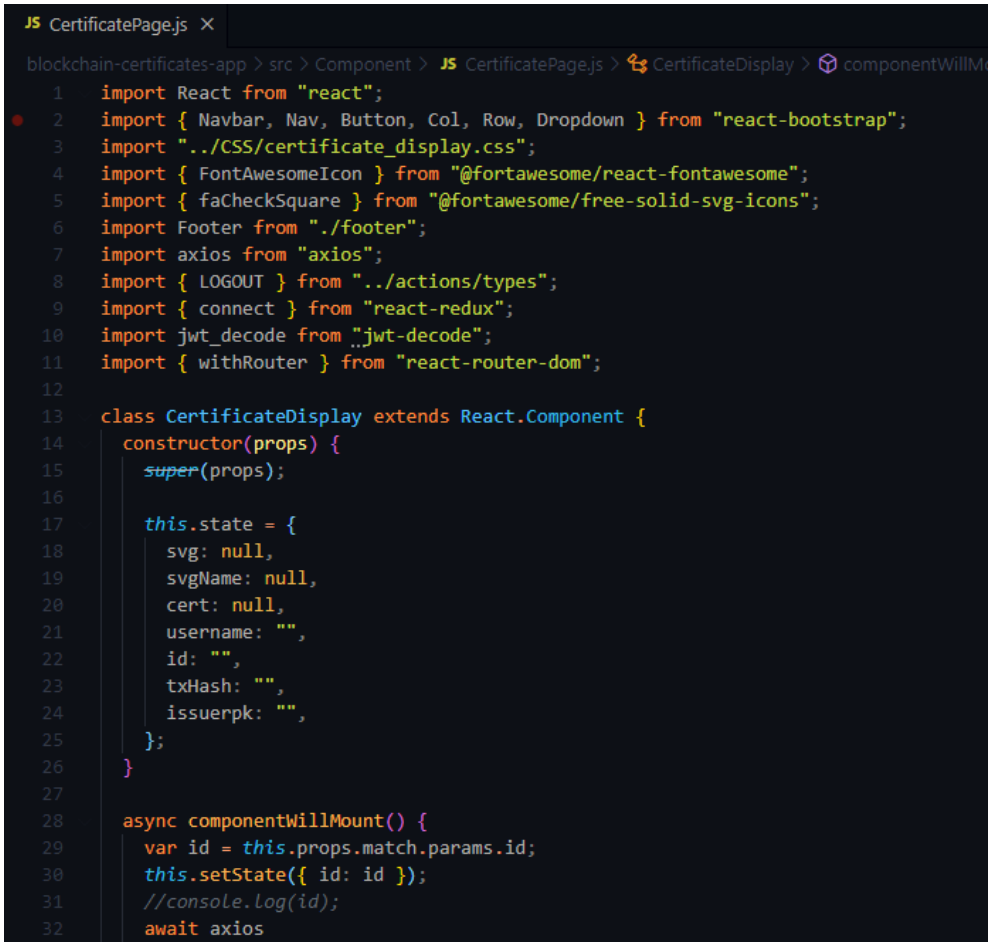


Figure 5.7: Program for students to view their certificates (a).

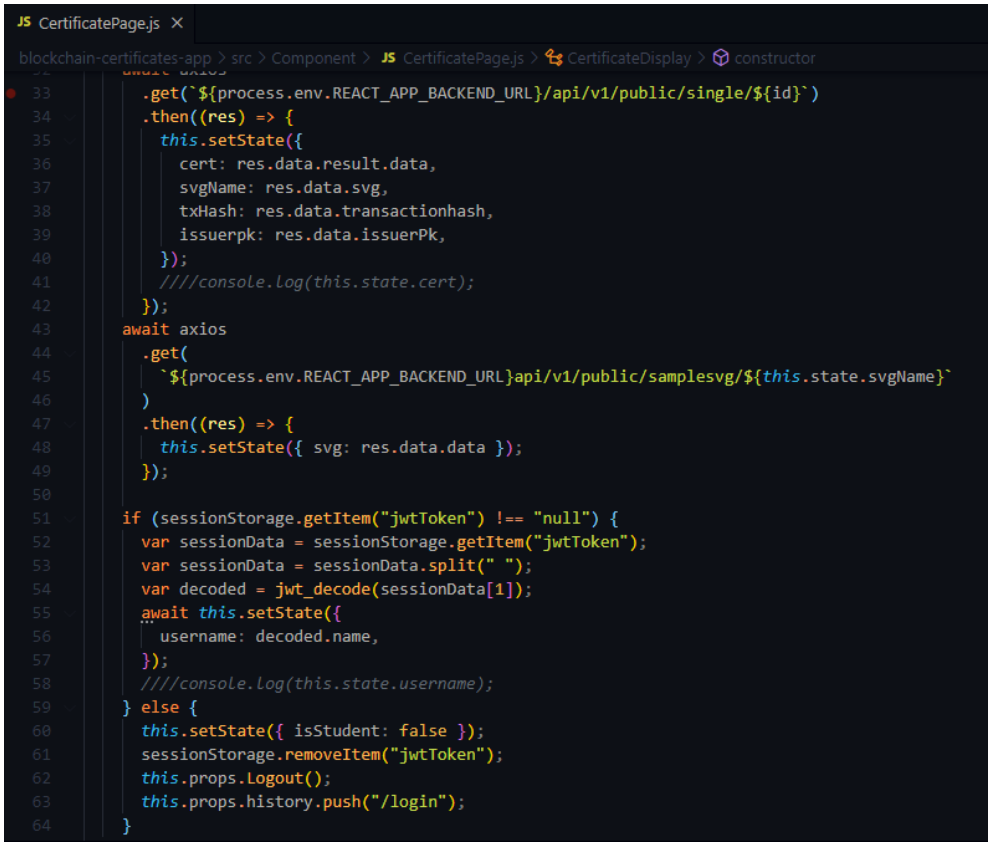


Figure 5.8: Program for students to view their certificates (b).

```
router.post("/addCerts", async (req, res, next) => {
  try {
    if (req.user.role !== "admin") {
      throw {
        statusCode: 400,
        customMessage: "not authorized!",
      };
    }
    console.log(req.user);
    var cert = req.body.cert; // json
    var svg = req.body.svg; // cert name/id.svg
    var query =
      "insert into certs(uploader,email,id,jsonstring,svg,cert_id) values";
    await cert.map((i, index) => {
      var token = crypto.randomBytes(16).toString("hex");
      query =
        query +
        `('${req.user.username}','${i.data.email}','${token}','${JSON.stringify(
          i
        )}','${svg}','${i.data.cert_id}'),`;
    });
    query = query.substring(0, query.length - 1);
    console.log(query);
    await pgp.query(query);

    res.status(200).json({
      message: "Data will be updated on the blockchain network shortly",
    });
  } catch (err) {
    console.log(err);
    next(err);
  }
});
```

Figure 5.9: Program to add certificate template in database.

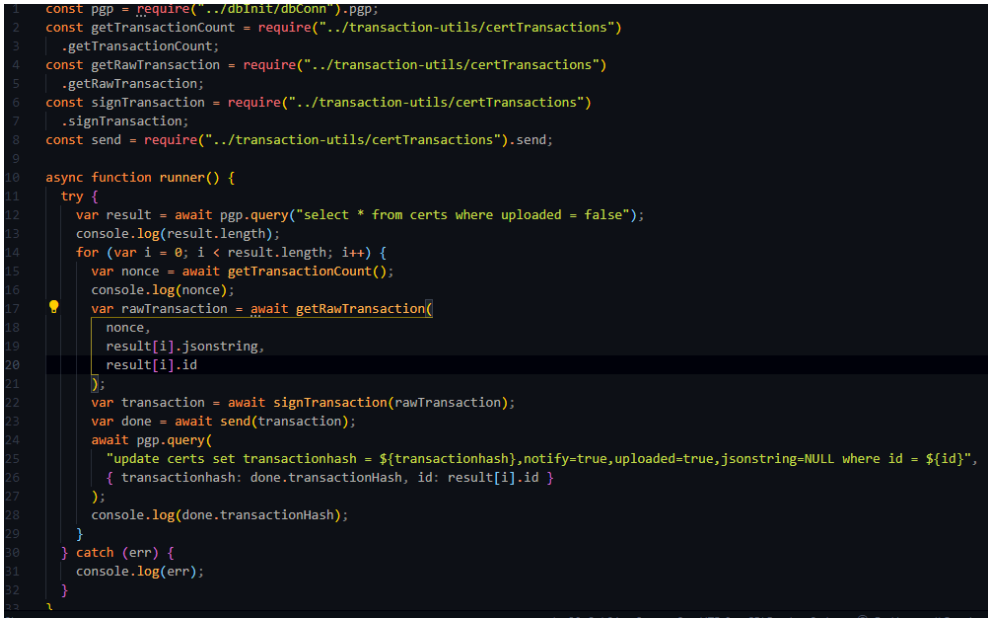
```
const Web3 = require("web3");
const infuraURL = config.get("infuraEndpoint");
const APIkey = config.get("infuraAPIkey");
const infura = `${infuraURL}/${APIkey}`;
const web3 = new Web3(new Web3.providers.HttpProvider(infura));
const abi = [
  {
    inputs: [
      {
        internalType: "string",
        name: "_keys",
        type: "string",
      },
      {
        internalType: "string",
        name: "_certs",
        type: "string",
      },
    ],
    name: "newCert",
    outputs: [],
    stateMutability: "nonpayable",
    type: "function",
  },
  {
    inputs: [],
    stateMutability: "nonpayable",
    type: "constructor",
  },
  {
    inputs: [
      {
        internalType: "string",
```

Figure 5.10: Smart Contract to set the structure of the blockchain network (a).

```
    {
      inputs: [
        {
          internalType: "string",
          name: "",
          type: "string",
        },
      ],
      name: "certificates",
      outputs: [
        {
          internalType: "string",
          name: "",
          type: "string",
        },
      ],
      stateMutability: "view",
      type: "function",
    },
  ];
  const contractAddr = config.get("contractAddr");

  var contract = new web3.eth.Contract(abi, contractAddr);
```

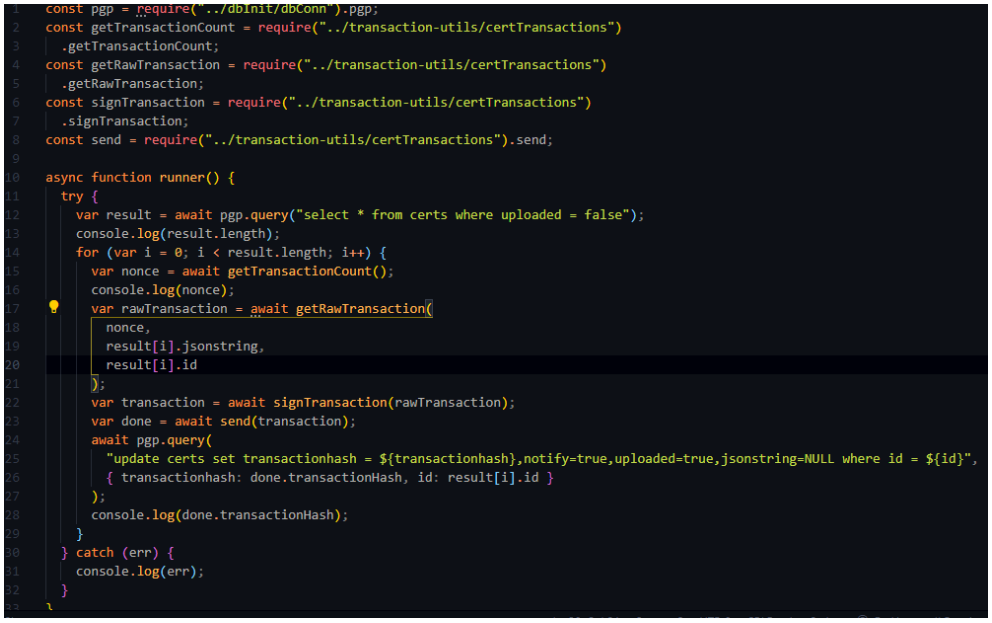
Figure 5.11: Smart Contract to set the structure of the blockchain network (b).



```
const pgp = require("../dbInit/dbConn").pgp;
const getTransactionCount = require("../transaction-utils/certTransactions")
  .getTransactionCount;
const getRawTransaction = require("../transaction-utils/certTransactions")
  .getRawTransaction;
const signTransaction = require("../transaction-utils/certTransactions")
  .signTransaction;
const send = require("../transaction-utils/certTransactions").send;

async function runner() {
  try {
    var result = await pgp.query("select * from certs where uploaded = false");
    console.log(result.length);
    for (var i = 0; i < result.length; i++) {
      var nonce = await getTransactionCount();
      console.log(nonce);
      var rawTransaction = await getRawTransaction(
        nonce,
        result[i].jsonstring,
        result[i].id
      );
      var transaction = await signTransaction(rawTransaction);
      var done = await send(transaction);
      await pgp.query(
        "update certs set transactionhash = ${transactionhash},notify=true,uploaded=true,jsonstring=NULL where id = ${id}",
        { transactionhash: done.transactionHash, id: result[i].id }
      );
      console.log(done.transactionHash);
    }
  } catch (err) {
    console.log(err);
  }
}
```

Figure 5.12: Program to upload data on the blockchain network.



```
const pgp = require("../dbInit/dbConn").pgp;
const getTransactionCount = require("../transaction-utils/certTransactions")
  .getTransactionCount;
const getRawTransaction = require("../transaction-utils/certTransactions")
  .getRawTransaction;
const signTransaction = require("../transaction-utils/certTransactions")
  .signTransaction;
const send = require("../transaction-utils/certTransactions").send;

async function runner() {
  try {
    var result = await pgp.query("select * from certs where uploaded = false");
    console.log(result.length);
    for (var i = 0; i < result.length; i++) {
      var nonce = await getTransactionCount();
      console.log(nonce);
      var rawTransaction = await getRawTransaction(
        nonce,
        result[i].jsonstring,
        result[i].id
      );
      var transaction = await signTransaction(rawTransaction);
      var done = await send(transaction);
      await pgp.query(
        "update certs set transactionhash = ${transactionhash},notify=true,uploaded=true,jsonstring=NULL where id = ${id}",
        { transactionhash: done.transactionHash, id: result[i].id }
      );
      console.log(done.transactionHash);
    }
  } catch (err) {
    console.log(err);
  }
}
```

Figure 5.13: Program to upload data from the database on to the blockchain network.

```
async function getTransactionCount() {
  return await web3.eth
    .getTransactionCount(addr)
    .then((result) => {
      return result;
    })
    .catch((error) => {
      console.log(error);
      next(error);
    });
}

function getRawTransaction(nonce, data, HashId) {
  console.log(nonce, data, HashId);
  var rawTransaction = {
    from: addr,
    gasPrice: web3.utils.toHex(20 * 1e9),
    gasLimit: web3.utils.toHex(300000),
    to: contractAddr,
    value: "0x0",
    data: contract.methods.newCert(HashId, data).encodeABI(),
    nonce: web3.utils.toHex(nonce),
  };
  return rawTransaction;
}

async function signTransaction(rawTransaction) {
  var transaction = new Tx(rawTransaction, {
    chain: "ropsten",
    hardfork: "petersburg",
  });
  //signing transaction with private key
```

Figure 5.14: Methods used to verify, get information and transaction hash number from the blockchain network.

```
const runner = require("../../runner/runner");
router.get("/runner", (req, res, next) => {
  try {
    console.log("called the runner");
    runner();
    res.status(200).json({ message: "called runner" });
  } catch (err) {
    next(err);
  }
});
```

Figure 5.15: Program used to upload data periodically on the blockchain network ensuring no data is being dropped or skipped.

```
const nodemailer = require("nodemailer");
const pgp = require("../dbInit/dbConn").pgp;
const config = require("config");
const transport = nodemailer.createTransport({
  service: "gmail",
  auth: {
    user: config.get("SMTPmail"),
    pass: config.get("SMTPpassword"),
  },
});
async function emailrunner() {
  try {
    var result = await pgp.query(
      "select * from certs where uploaded = true and notify = true limit 500"
    );

    for (var i = 0; i < result.length; i++) {
      var mailOptions = {
        from: config.get("SMTPmail"),
        to: result[i].email,
        subject: "",
        text: "result[i].id",
        html: `<a href="https://ropsten.etherscan.io/tx/${result[i].transactionhash}><h3>click here to view the transaction</h3></a>`
      };
      var data = await transport.sendMail(mailOptions);
      await pgp.query("update certs set notify=false where id = ${id}", {
        id: result[i].id,
      });
      console.log(data);
    }
  } catch (err) {
    console.log(err);
  }
```

Figure 5.16: Program used to send the transaction value or the unique key of the certificate to the appropriate owner of the certificate once it has been uploaded on the blockchain network.

Chapter 6

Testing

Test No	Test Name	Expected Result	Actual Result
1	Register new user	Register user to the database	Registered successfully
2	Login as admin	Open admin dashboard	Logged in successfully
3	Login as student	Open student dashboard	Logged in successfully
4	Upload SVG template of certificate	Save SVG template with proper serial number into the database	Saved successfully
5	Upload CSV data onto the blockchain network	Save CSV data on blockchain network with template serial number	Saved successfully
6	Receive mail	Mail to be received when certificate is validated	Mail received successfully
7	Retrieve data from blockchain	Retrieve appropriate certificate data requested by student	Retrieved data successfully
8	Mapping the retrieved data	Map the data retrieved from the blockchain on the certificate template	Mapping of data successfully

Table 6.1: Testing different modules of the system

Chapter 7

Result

The following images are the results/output screenshots of our finished application "Educhain".

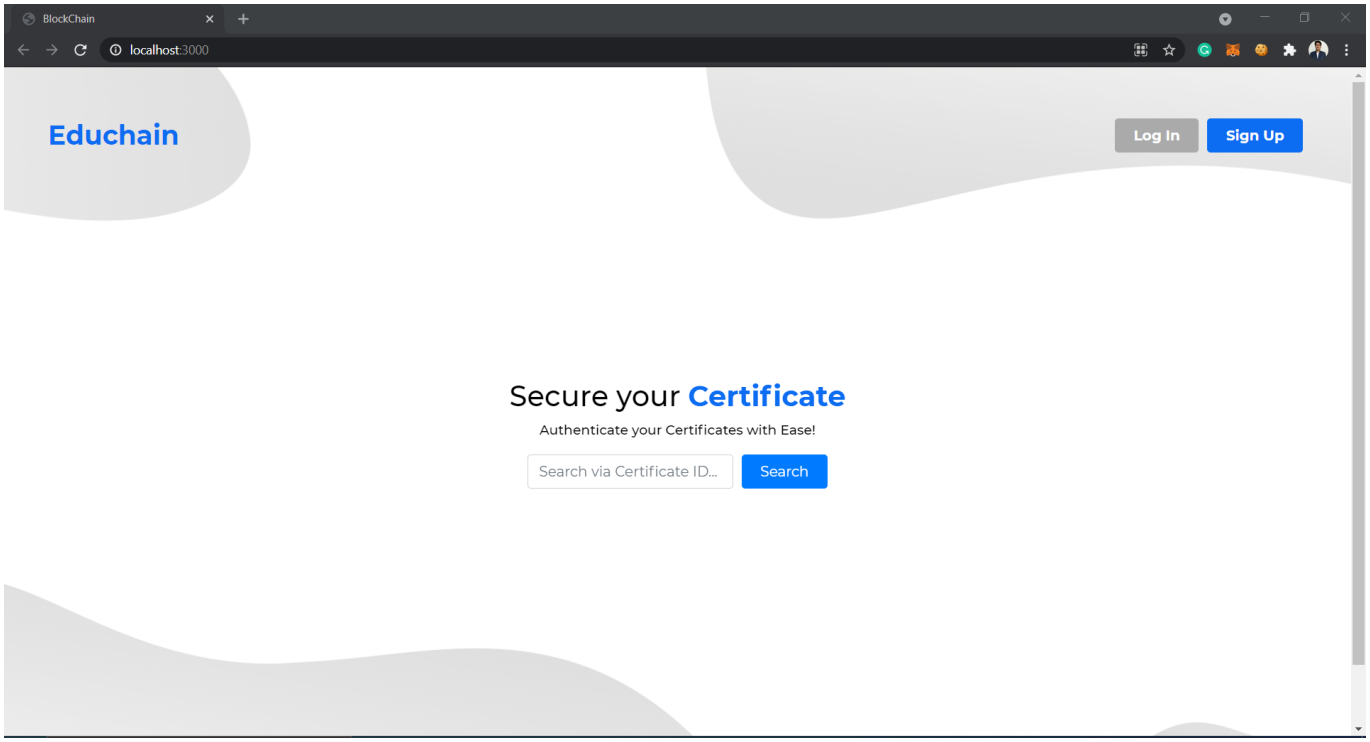


Figure 7.1: Home screen.

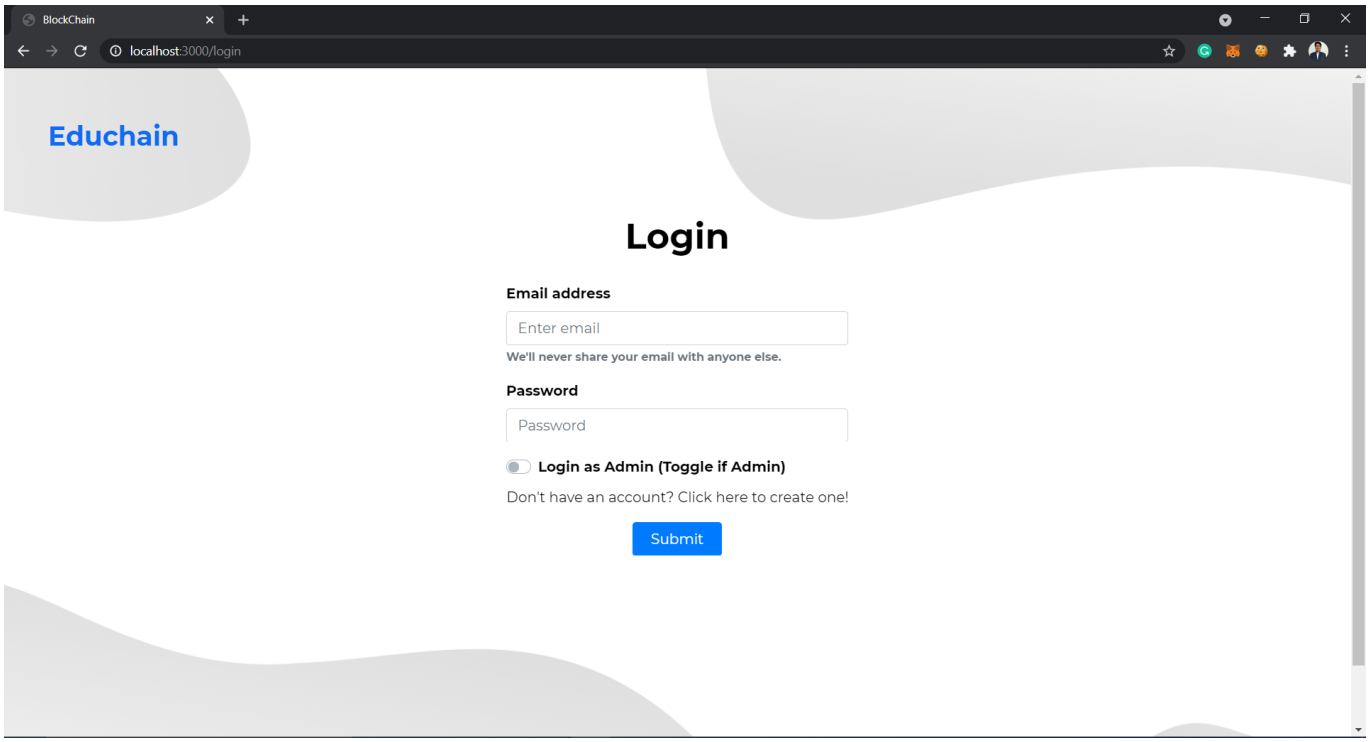


Figure 7.2: Login page.

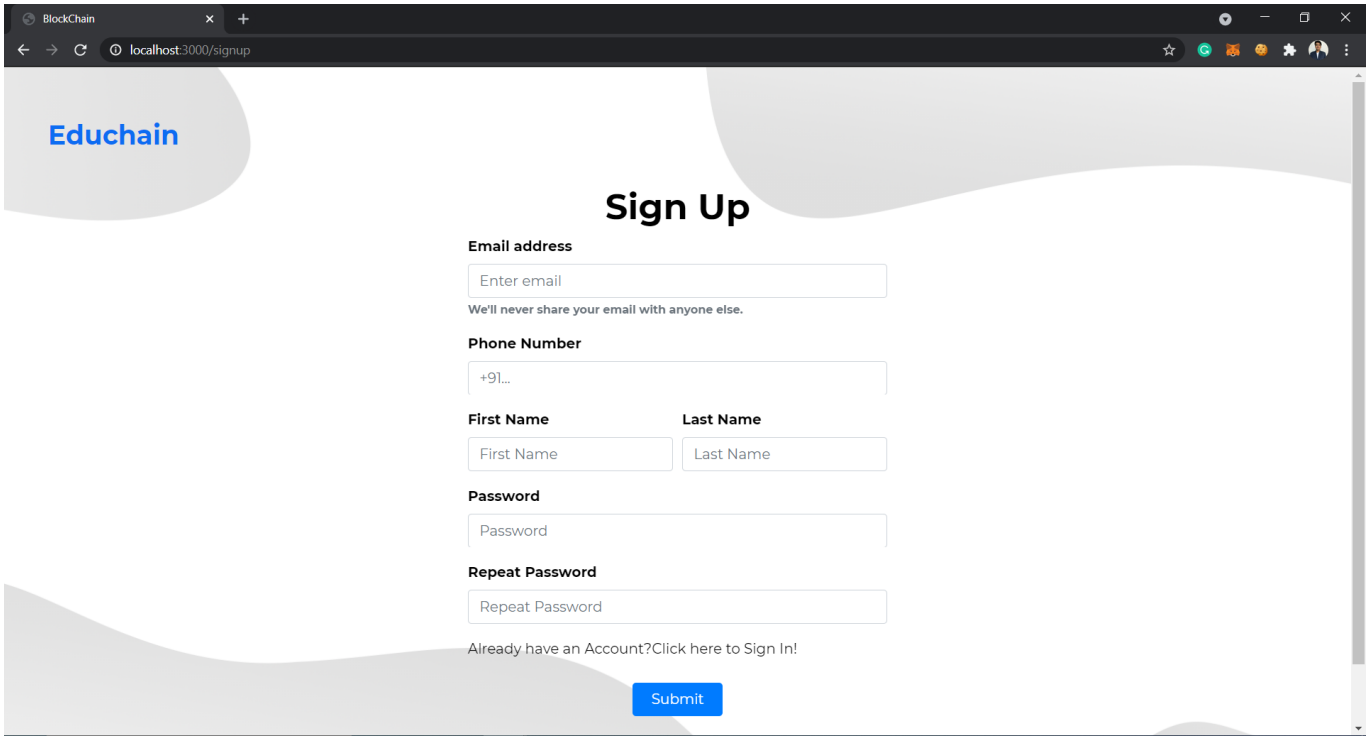


Figure 7.3: Registration page.

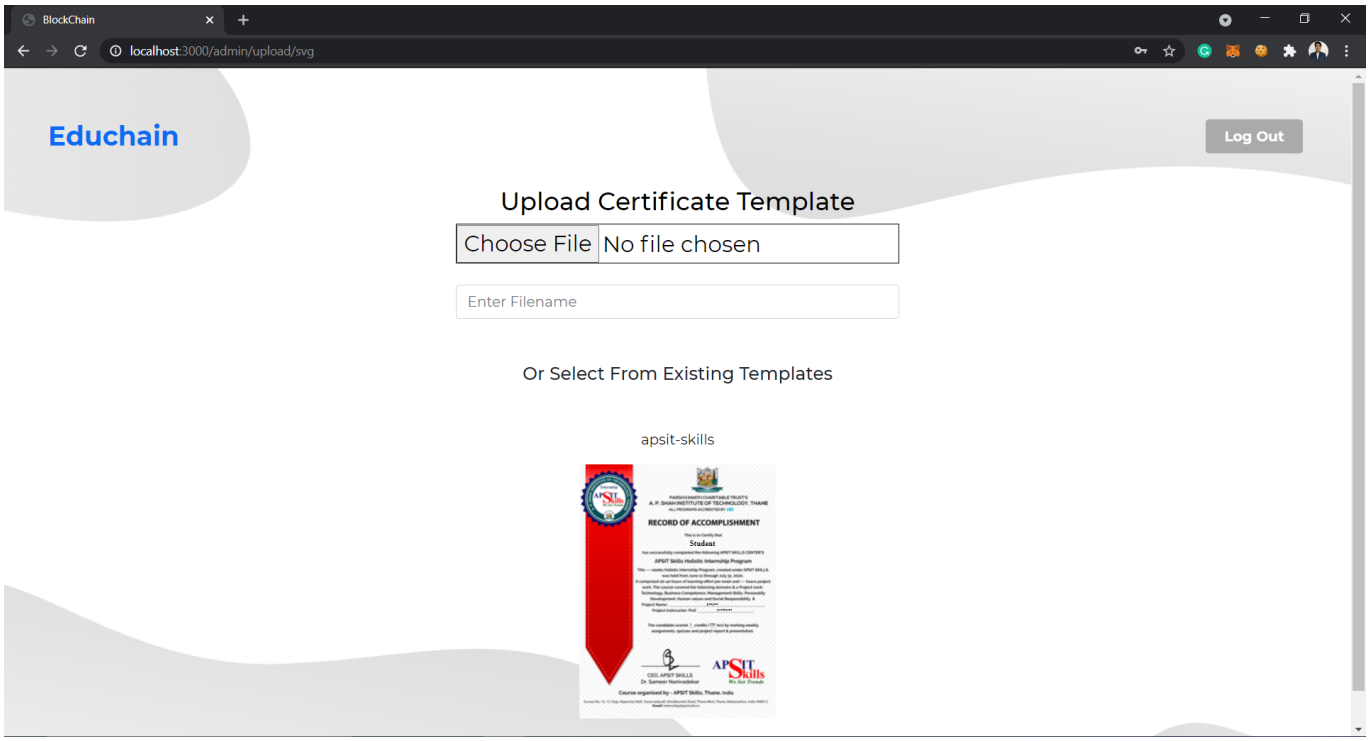


Figure 7.4: Admin portal to add certificate template in SVG format.

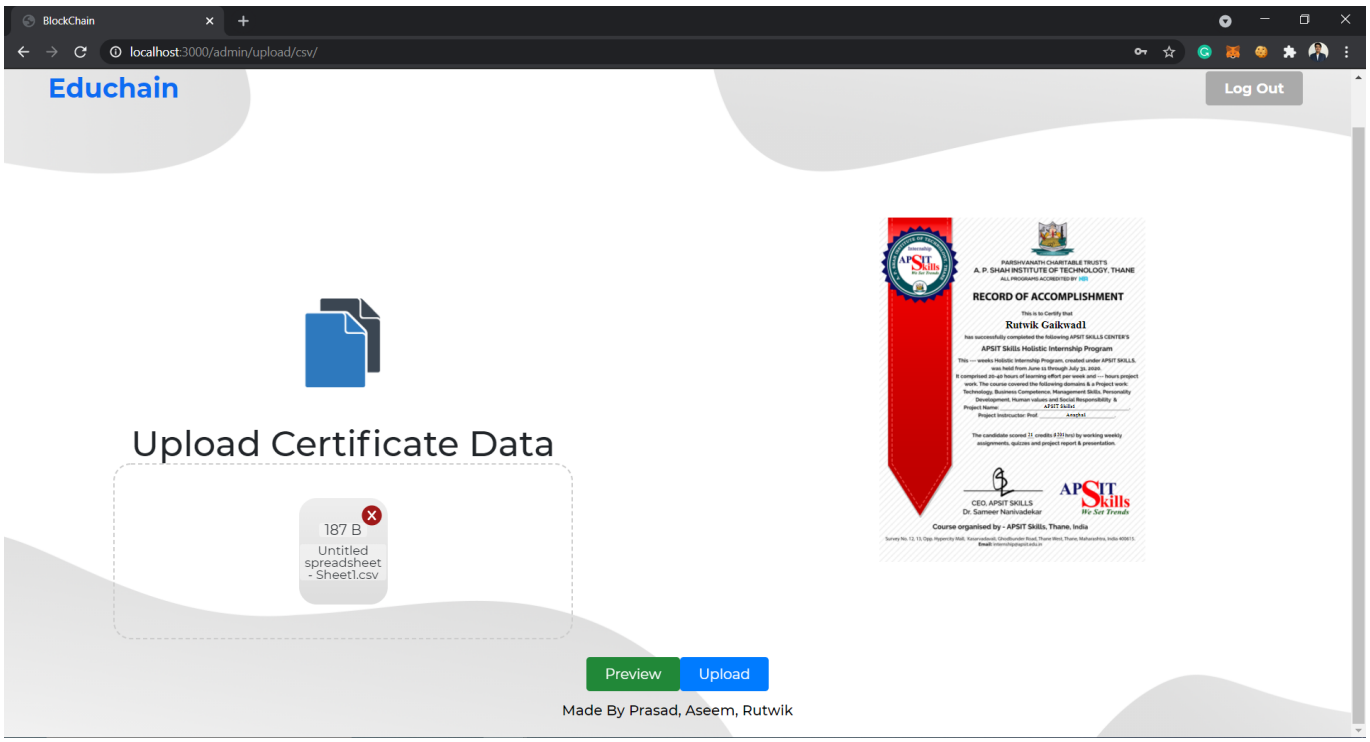


Figure 7.5: Admin portal to add CSV file for the selected template with the preview option.

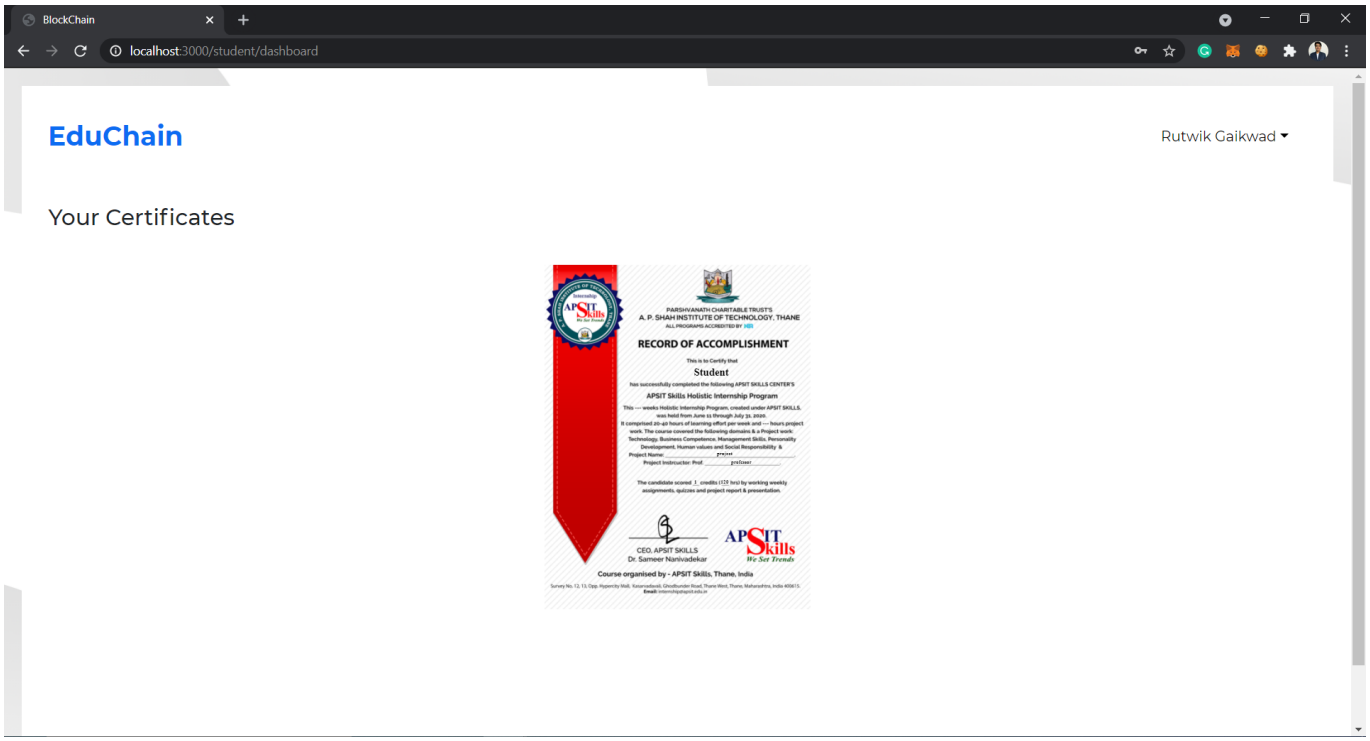


Figure 7.6: Dashboard for Students to view all Certificates

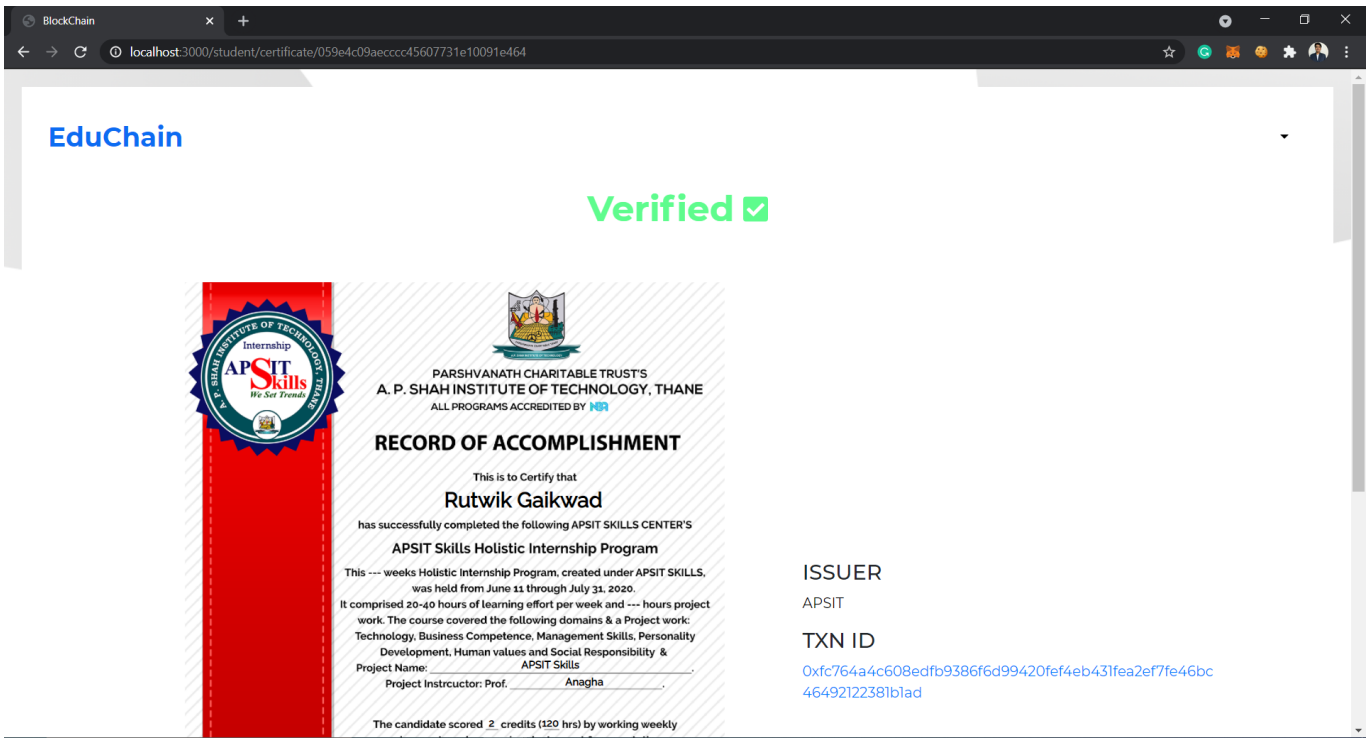


Figure 7.7: Display of Individual Certificate with verification from BlockChain

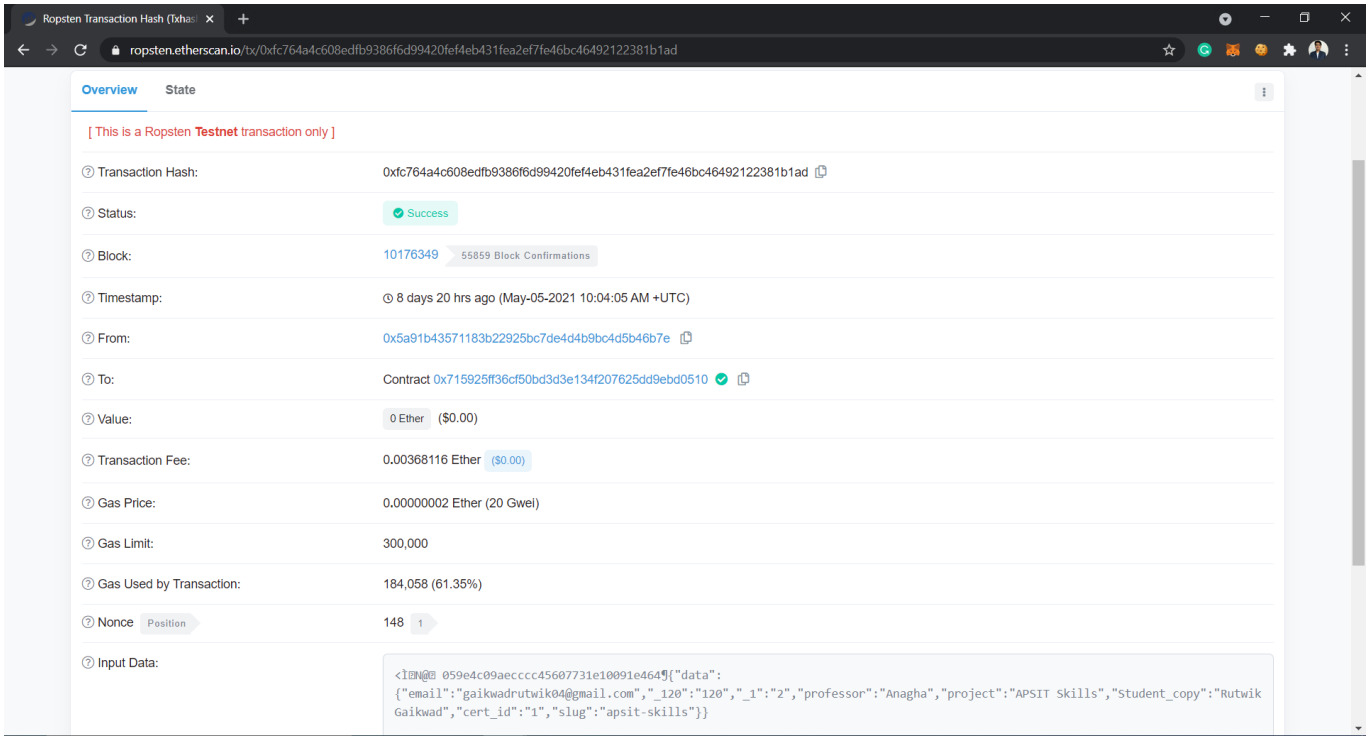


Figure 7.8: Verify transaction done on BlockChain

Chapter 8

Conclusions and Future Scope

8.0.1 Conclusion

In this paper, we have successfully proposed a system where blockchain technology can be used to store and retrieve certificate data. The project will help companies issue certificates securely through Blockchain and can be verified by anyone with the unique link/code to each certificate. The system uses the concept of SVG templates for the certificates which would be stored on a local server and to store data over the blockchain for secure and reliable storage. This will minimize the cost of storing the entire certificate on the blockchain network. Storing only the data of the certificate will minimize cost and thereby turn out to be cost-efficient. The only drawback of this system is that the template of the certificate needs to be properly created with the SVG's text area ID to match with the header of the CSV file. Only a properly crafted SVG and CSV pair will result in proper certificate generation through Blockchain.

8.0.2 Future Scope

1. Currently, the certificate template which is stored on local file storage is the weakest link in the system. The template relies on the security of the File System used. The use of IPFS - InterPlanetary File System can secure the certificate template stored thereby adding to the security of the system.

2. IPFS has the capacity to store files over Blockchain allowing secure storage and retrieval of the certificate template.

3. The system can be further extended to store other online documents of importance to ensure the integrity of data and documents being stored securely.

Bibliography

[1] UK: NARIC Survey: universities and education fraud – 75% cannot spot a fake certificate - Feb 2020, https://uknaric.org/2020/02/26/survey-universities-and-education-fraud-75-cannot-spot-a-fake-certificate/

[2] BBC File on 4 exposes a multi-million pound global trade in fake diplomas - Jan 2018, https://www.bbc.co.uk/programmes/b09ly731

[3] Blockchain, https://en.wikipedia.org/wiki/Blockchain.

[4] Smart Contracts, https://www.ibm.com/blogs/blockchain/2018/07/what-are-smart-contracts-on-blockchain/

[5] Scalable Vector Graphics, https://en.wikipedia.org/wiki/Scalable_Vector_Graphics

[6] Shanmuga Priya R, Swetha N. Online Certificate Validation Using Blockchain. Published in Int. Jnl. Of Advanced Networking Applications (IJANA)

[7] Nitin Kumavat, Swapnil Mengade, Dishant Desai, JesalVarolia (2019). Certificate Verification System using Blockchain. International Journal for Research in Applied Science Engineering Technology (IJRASET).

[8] Oliver Miquel, Moreno Joan,Prieto Gerson, Ben tez David (2018). Using blockchain as a tool for tracking and verification of official degrees: business model 29th European Regional Conference of the International Telecommunications Society (ITS): "Towards a Digital Future: Turning Technology into Markets?", Trento, Italy, 1st - 4th August 2018.

[9] K. Kuvshinov, I. Nikiforov, J. Mostovoy, and D. Mukhutdinov, "Disciplina: Blockchain for Education," pp. 1–17, 2018.

[10] D.T.T. Anh, M. Zhang, B.C. Ooi, and G. Chen, "Untangling Blockchain: A Data Processing View of Blockchain Systems," IEEE Trans. Knowl. Data Eng., vol. 4347, no. c, pp. 1–20, 2018.

[11] Kumar, NM Saravana. quot;Implementation of artificial intelligence in imparting education and evaluating student performance." Journal of Artificial Intelligence 1, no. 01 (2019): 1-9.

[12] Smys, S., Joy Iong Zong Chen, and Subarna Shakya. quot;Survey on Neural Network Architectures with Deep Learning." Journal of Soft Computing Paradigm (JSCP) 2, no. 03 (2020): 186-194

Acknowledgement

We have great pleasure in presenting the report on **Online Certificate Generation Verification using Blockchain Framework** We take this opportunity to express our sincere thanks towards our guide **Prof. Kiran Deshpande** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Prasad Jadhav:
17104003:

Rutwik Gaikwad:
17104074:

Aseem Godambe:
17104058:

Publication

Paper entitled **"Online Certificate Generation & Verification using Blockchain Framework"** is presented at **"ICSCS 2021 : International Conference on Soft Computing for Security Applications"** by **Prasad Jadhav**, **Rutwik Gaikwad**, **Aseem Godambe** and **Kiran Deshpande**.