A

Major Project

On

A SYSTEM FOR ACADEMIC CERTIFICATES VERIFICATION USING BLOCKCHAIN

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

T. Sai Charitha (187R1A0556)

Kandakatla Anirudh Baba (187R1A0526)

Under the Guidance of

NUTHANAKANTI BHASKAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**A SYSTEM FOR ACADEMIC CERTIFICATES VERIFICATION USING BLOCKCHAIN**" being submitted by **T. SAI CHARITHA (187R1A0556), KANDAKATLA ANIRUDH BABA (187R1A0526)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Nuthanakanti Bhaskar
Assistant Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on_____

ACKNOWLEGDEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Nuthanakanti Bhaskar** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work . The blessing, help and guidance given by his shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Mr. A. Uday Kiran, Mr. J. Narasimha Rao, Dr. T. S.Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar** for their cordial support, valuable information and guidance, which helped us incompleting this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally,we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge all those who gave support directly and indirectly in completion of this project.

T. SAI CHARITHA (187R1A0556)

KANDAKATLA ANIRUDH BABA (187R1A0526)

ABSTRACT

The DApp (Decentralised application) being developed enables easy verification of credentials by storing the certificates on Ethereum blockchain network using IPFS (Inter Planetary File System) which is a distributed file system, thereby making the information stored immutable and secure. The website is being developed in three phases. In the first phase, the college enrolls students and uploads their credentials on the Ethereum blockchain. In the second phase, students can view their credentials and access requests sent by companies. In the third phase, companies can send access requests to students whose credentials they want to verify. Once the students accept the access requests, companies can view and verify the certificates.

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

1.INTRODUCTION

1.1 DECENTRALISED APPLICATIONS

A decentralized application (dApp) is a type of distributed open source software application that runs on a peer-to-peer (P2P) blockchain network rather than on a single computer. DApps are visibly similar to other software applications that are supported on a website or mobile device but are P2P supported. DApps are built on a decentralized network that is supported by a blockchain distributed ledger. The use of blockchain enables a dApp to process data through distributed networks and to execute transactions. dApps are also often built using the Ethereum platform. Distributed ledger technologies like the Ethereum blockchain have helped popularize dApps. The major advantages of dApps are that they are always accessible and do not have a single point of failure. While a traditional application is supported by centralized servers and database, a dApp is supported by a smart contract that is stored on a blockchain. Ethereum is the most popular blockchain for running smart contracts. Smart contracts enforce rules defined in the code and mediate transactions. Since a smart contract consists of the back-end only and is often just a small part of the whole dApp, creating a decentralized app on a smart contract system requires combining several smart contracts and employing third-party systems for the front-end.

1.1.1 ATTRIBUTES OF DAPPS

- They are open source**.** All required changes are decided upon by a consensus.
- They provide decentralized storage**.** Data is stored on decentralized blocks.
- They offer cryptographic Decentralized blocks of data are validated and proven true.

1.1.2 WEB 2.0 AND WEB 3.0

Web 2.0 and Web 3.0 refer to successive iterations of the web, compared with the original Web 1.0 of the 1990s and early 2000s. Web 2.0 is the current version of the internet (a term often used interchangeably with the web) with which we are all familiar, while Web 3.0 represents its next phase. Defining features of Web 3.0 include decentralization; trustlessness and permissionlessness; artificial intelligence (AI) and machine learning; and

connectivity and ubiquity. n Web 2.0, computers use HTTP in the form of unique web addresses to find information, which is stored at a fixed location, generally on a single server. With Web 3.0, because information would be found based on its content, it could be stored in multiple locations simultaneously and hence be decentralized. This would break down the massive databases currently held by internet giants like Meta and Google and would hand greater control to users.

1.1.3 ETHEREUM BLOCKCHAIN

Ethereum offers an extremely flexible platform on which to build decentralized applications using the native Solidity scripting language and Ethereum Virtual Machine. Decentralized application developers who deploy smart contracts on Ethereum benefit from the rich ecosystem of developer tooling and established best practices that have come with the maturity of the protocol. This maturity also extends into the quality of user-experience for the average user of Ethereum applications, with wallets like MetaMask, Argent, Rainbow and more offering simple interfaces through which to interact with the Ethereum blockchain and smart contracts deployed there.

1.2 EXISTING CERTIFICATE VERIFICATION PROCESS

Marks memos are issued directly to the students as a hard copy. There is no digitalized way to verify the certificate. Once the certificate is distributed among the students, there will be no connection between students, university, and the certificate. There is no platform to store the certificate safely and verify them when required. Therefore fake graduation degree certificates are created to get backdoor jobs. In industries, once an employee is hired, they require a background check of the educational details of the employee, and this verification is done just manually by their HR team or by some third party. There may be a delay in the process and a chance to manage the concerned section personnel of the university or college who receive the verification calls. It is even difficult to distinguish the fake and original degrees if the master register has already been tampered. Some universities store certificates in digital form but are also in a centralized network where there is a chance of tampering the certificate.

1.3 PROJECT SCOPE

This project is titled as “A Blockchain based verification system for academic certificates”. This software provides facility to upload certificates on Ethereum blockchain. Storing any information on blockchain makes it immutable and this immutability feature of blockchain is what makes it secure. Certificates can be uploaded on the blockchain so nobody except the college authority can change or tamper with the certificate.

1.4 PROJECT PURPOSE

This has been developed enhance the document verification process using blockchain technology. To make the data more secure and safe, everything needs to be digitized with the principle of confidentiality, reliability and availability. All the above can be achieved with blockchain technology. The data in blockchain cannot be changed under realistic conditions. Even if the data is changed, it just takes a second to let us know about the ampering. Through blockchain, the issue of tampering with the certificates is solved and also the time taken for the companies to validate the certificate is very less.

1.5 PROJECT FEATURES

The participants in the project are college, student and company. The main features of the project are enrolling the student and uploading the certificate onto the blockchain, both of which are done by college authorities. Students can view their certificates by logging in to their account. Companies can view the certificates by sending an access request to students whose certificate they want to access. Once the student accepts the access request, company can view that particular student’s credential.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A detailed study is made and the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

2.2 EXISTING SYSTEM

In the existing system, There is no digitalized way to verify the certificate.Some universities store certificates in digital form but are also in a centralized network where there is a chance of tampering the certificate. This may increase the cases of fraud since there is no means of security and integrity of the data both in manual and in digital form.The main reasons behind this problem are the lack of timestamp facility and method of storing data at a central storage.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

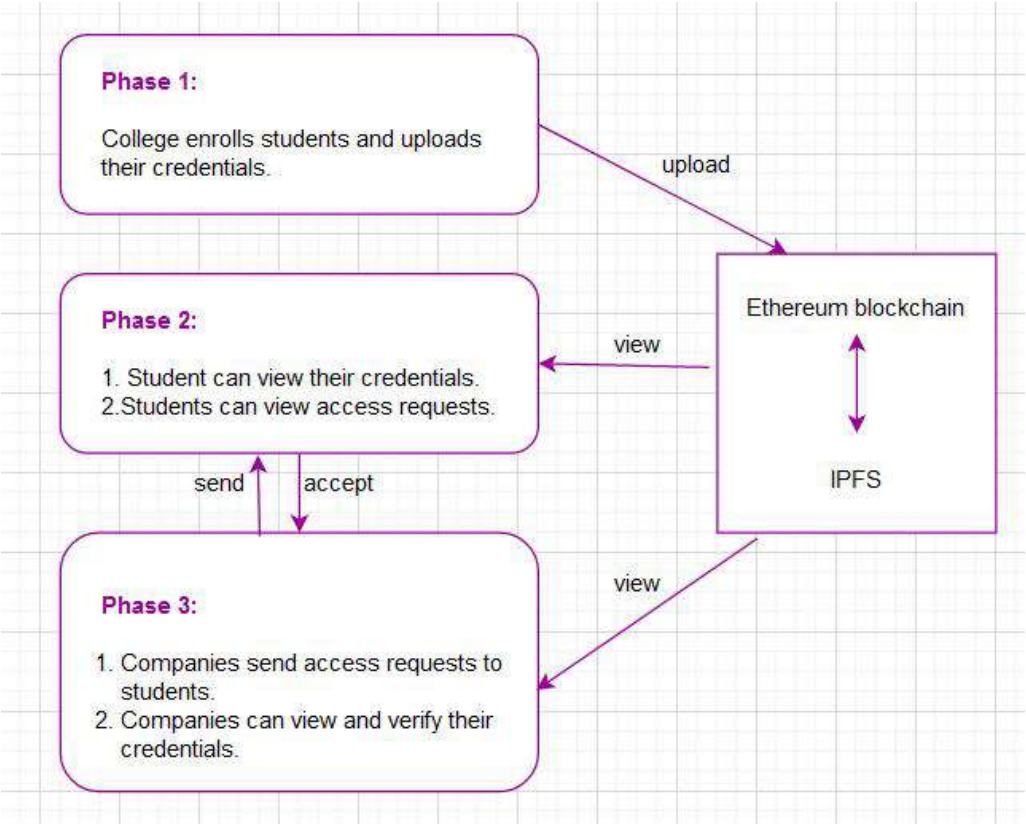
• There is no digitalized way to verify the certificates.

• Delay in the employer verification process.

• Increases the chances of fraud.

2.3 PROPOSED SYSTEM

In this proposed system, we provide a platform to store and verify the student credentials using blockchain technology.With the help of the unique certificate ID, student can verify the certificate and also the company can verify whether the certificate provided by the student is authorized or not. As the blockchain is distributed in nature and is popularly known as a distributed ledger, it is not easy to tamper the data stored in a block.



2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design. It has got following features

- No one can tamper or create any fake degrees.
- Immutable and distributed nature of already created blocks of data in the proposed credentials blockchain.
- Convenient system for users.
- Employer verification becomes easy and seamless.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that he project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel i3 or above.
- Hard disk : 10GB
- RAM : 2GB or above

2.5.2 SOFTWARE REQUIREMENTS:

Software requirements specifies the logical characteristics of each interface and software components of this system. The following are some software requirements.

- Operating system : Windows 7 or above
- Frontend : HTML, CSS and JavaScript
- Backend : Solidity, Node js, Truffle
- Code Editor : Visual Studio Code

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCITECTURE

As shown in Figure 3.1, the project architecture shows what pages the website is going to have and what functionalities it includes.

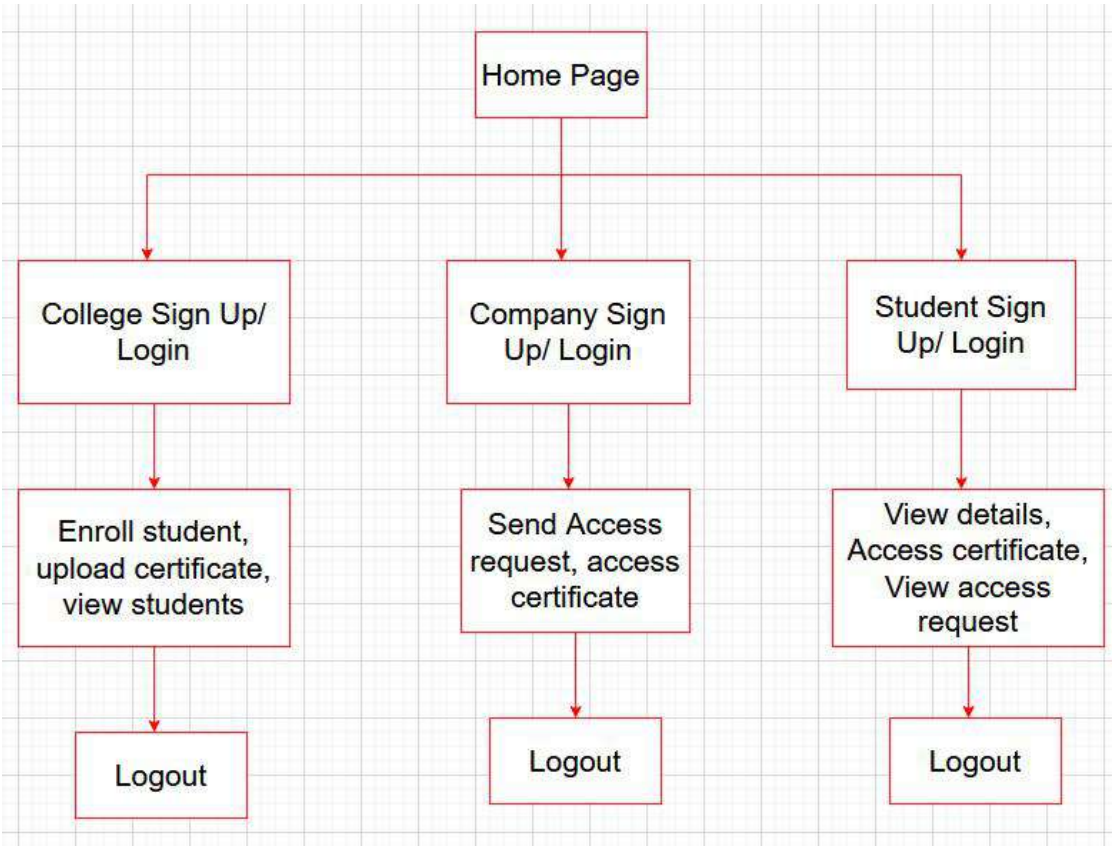


Figure 3.1: Project architecture

3.2 DESCRIPTION

Login/ Sign-up: New users have to create an account using the sign-up page. Users have to provide an email address with which they want to register and set a password.

Home Page: It consists of three accounts to choose: college, student or company.

Enroll Student: College can enroll students on this page by assigning unique ids to each student.

Upload Certificate: College can upload certificate on the blockchain and need to pay some gas fees.

Send access request: Companies can send access request to student whose certificate they want to verify.

Accept access requests: Student can accept or reject the access request sent by companies.

3.3 USE CASE DIAGRAM

In the use case diagram, we have 3 actors, college, student and company. College can enroll students and upload their certificates as shown in the figure 3.2. Student can view their certificate and also view access requests. Companies can send access requests and view certificates of students who have approved access requests.

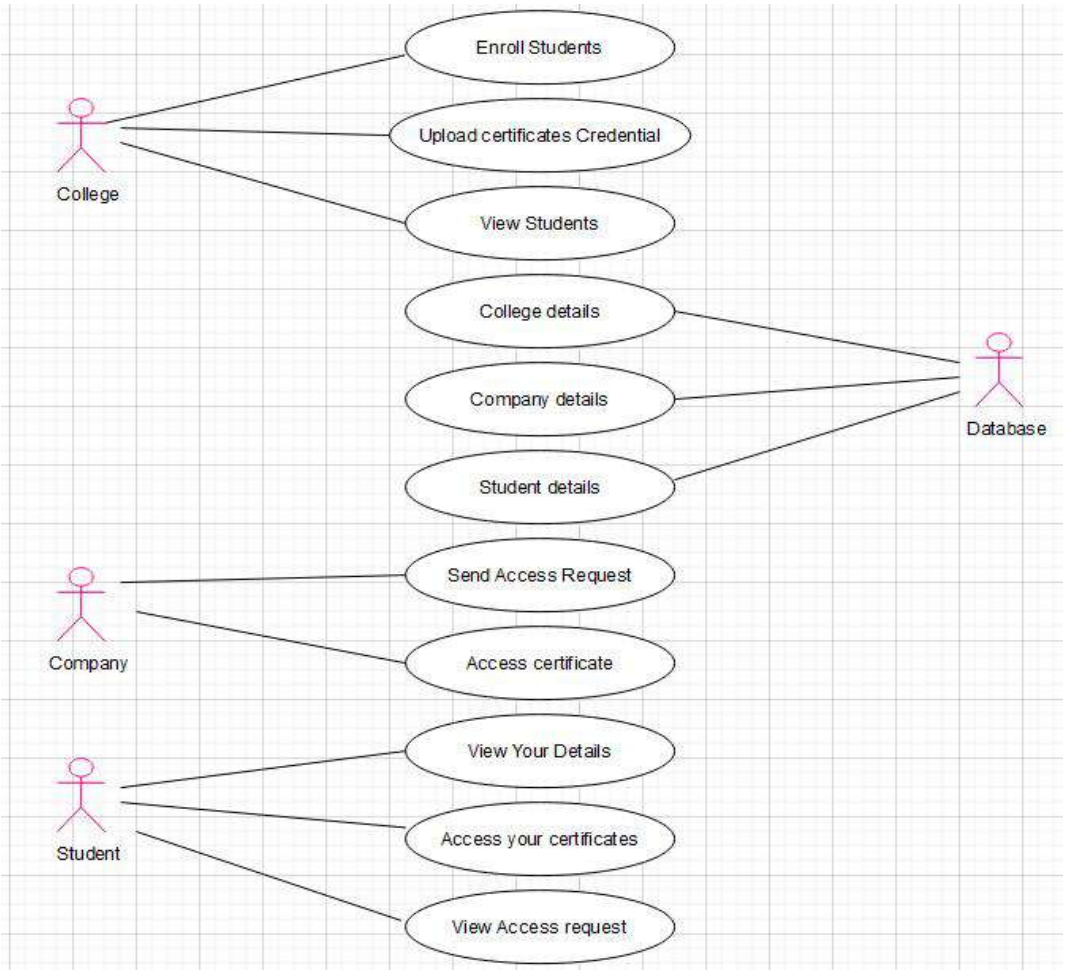


Figure 3.2: Use case diagram

3.4 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Here, college enrolls the student and uploads the certificate and students account gets updated and they can view the uploaded certificate as shown in the figure 3.3. Similarly, when a student accepts access request, the companies can view the certificate.

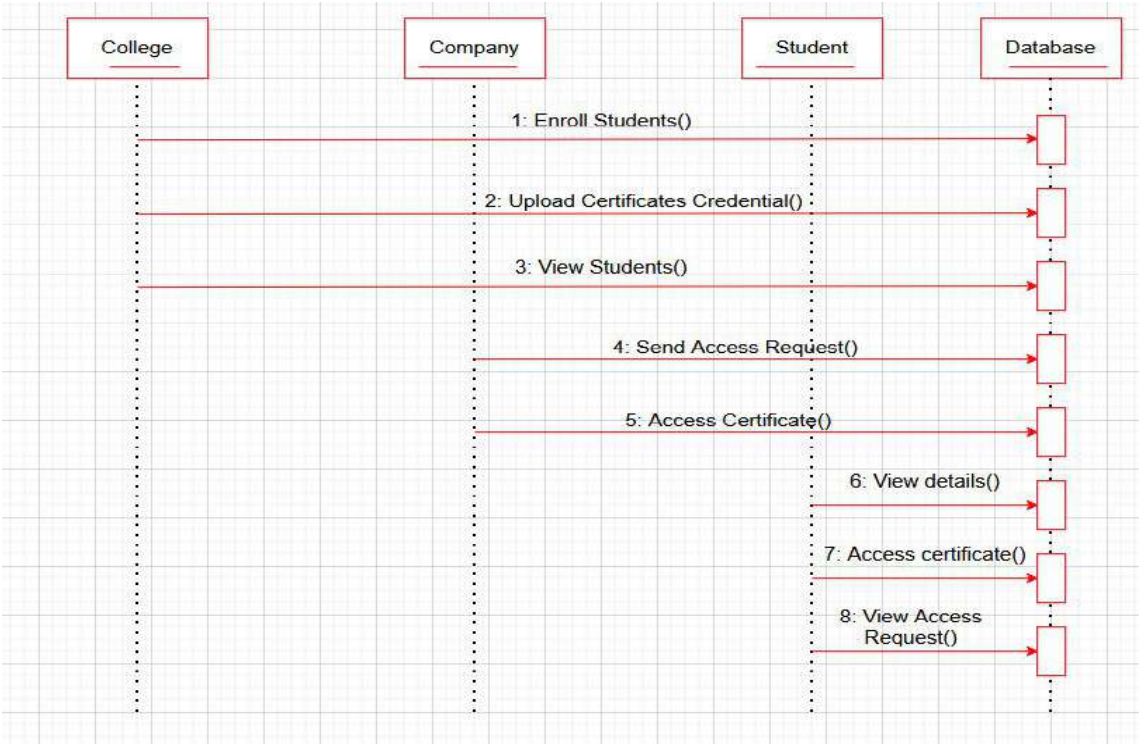


Figure 3.3: Sequence Diagram

3.5 ACTIVITY DIAGRAM

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. As shown in the figure 3.4, first, the user credentials are verified. If they are valid then home page is displayed otherwise the login page reloads asking to enter valid credentials.

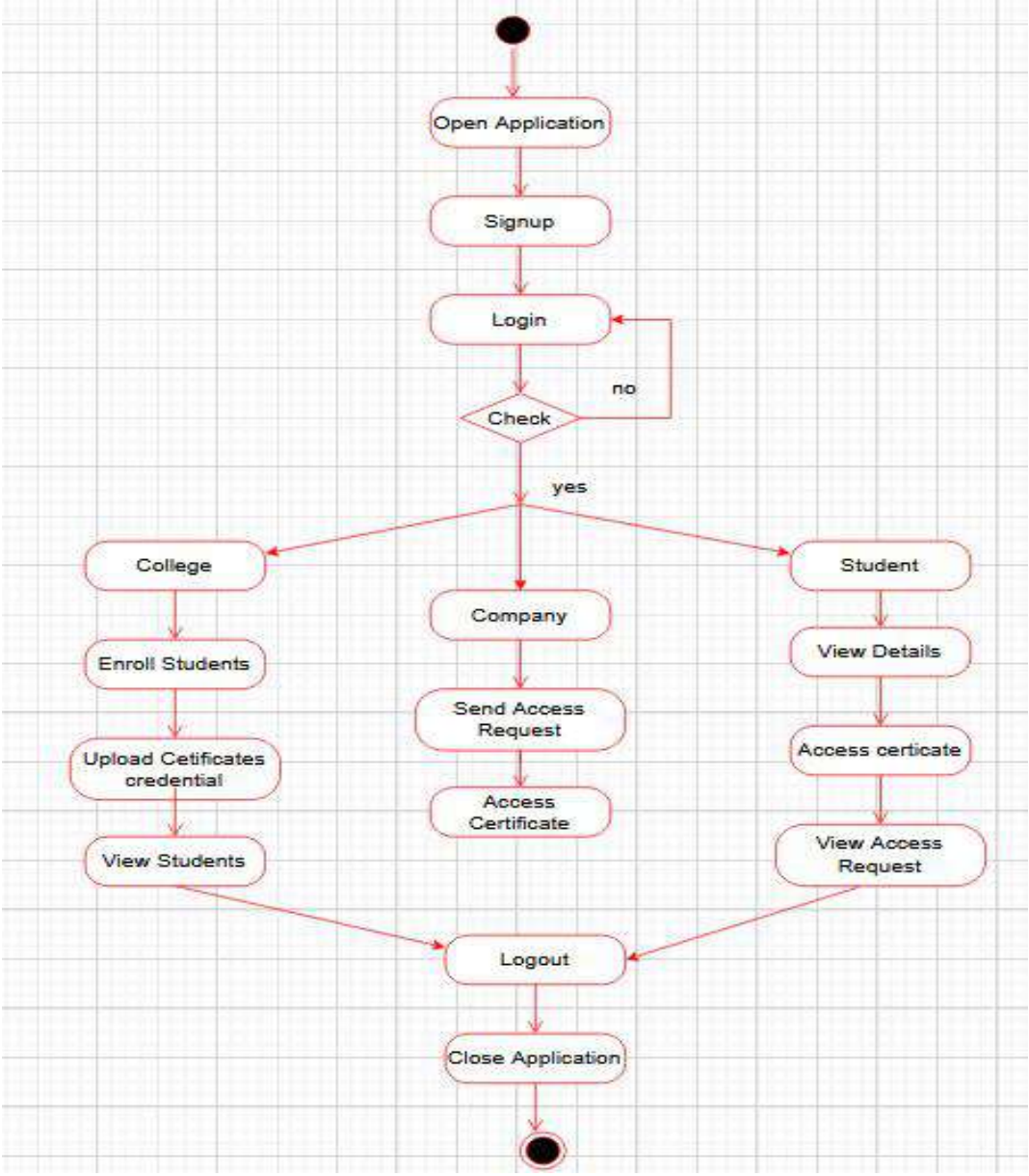
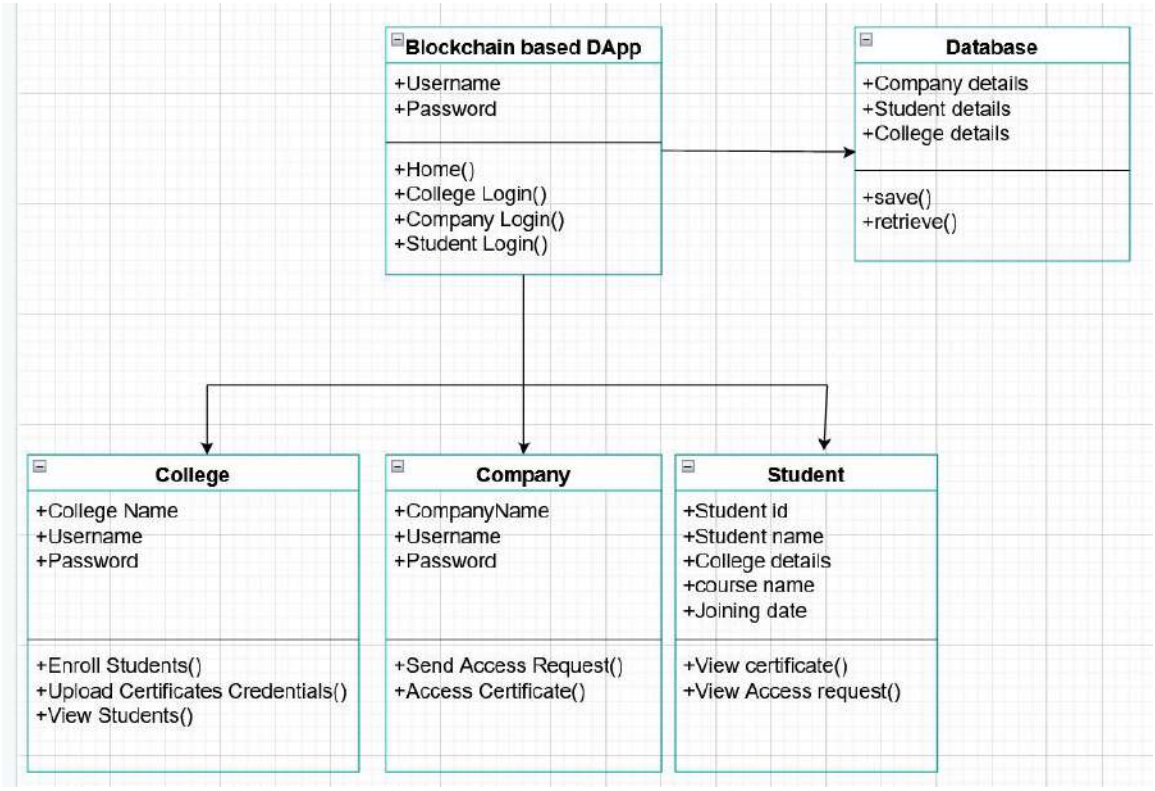


Figure 3.4: Activity Diagram

3.6 CLASS DIAGRAM

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram for all three profiles, college, student and company is as shown in figure 3.5.



Screenshot 3.5: Class Diagram

4.IMPLEMENTATION

4. IMPLEMENTATION

4.1 TOOLS AND TECHNOLOGIES

The proposed system is implemented and tested by using the following softwares: JavaScript, Truffle, Solidity, Ganache, Ethereum, and Chrome extension Metamask. Ganache is part of the Truffle ecosystem. Ganache is used for the development of DAPP (distributed application, a blockchain) and once it is developed and tested on ganache, it can be deployed on ethereum client like geth or parity. Truffle helps to develop, test, and deploy the DAPP. Metamask is one of the digital currency wallets to store and transact on ethereum using ethereum based tokens. In the front-end, we have used HTML, CSS and JavaScript. Back-end is implemented using JavaScript library node.js. The Ethereum smart contract is implemented using Solidity. The smart contract is compiled, deployed and tested using Truffle. Ganache is used for setting up a local Ethereum blockchain for testing the smart contracts. Ganache provides 10 accounts with fake ether which can be used for transactions while testing. To connect front-end and back-end with smart contract, we have used web.js which is a JavaScript library to build web 3.0 applications.

4.2 IMPLEMENTATION

In this system, the marks are stored as an image. A consortium of colleges and universities manages this blockchain.Students who want to store their academic details on this authorized platform should approach the consortium.This distributed ledger (blockchain) keeps track of every academic detail of the student from X class to graduation/post-graduation.Only the authorized users can add marks into the blockchain.The credentials of the students are added along with roll number, name, marks.All the generated certificates are attached with an individual certificate ID and this certificate ID is the unique ID used for verification.If anyone wants to check how many certificates are generated on a particular roll number, then by entering roll number, a report will show all the certificates generated on the particular name and roll number.

4.3 SAMPLE CODE

col-enroll.js

```
import Web3 from 'web3';
import Cert from '../../build/contracts/Cert.json';
let web3;
let cert;
const initWeb3 = () => {
  return new Promise((resolve, reject) => {
    if (typeof window.ethereum !== 'undefined') {
      const web3 = new Web3(window.ethereum);
      window.ethereum
        .enable()
        .then(() => {
          resolve(new Web3(window.ethereum));
        })
        .catch(e => {
          reject(e);
        });
      return;
    }
    if (typeof window.web3 !== 'undefined') {
      return resolve(
        new Web3(new Web3.providers.HttpProvider('http://127.0.0.1:7545'))
      );
    }
    resolve(new Web3('http://localhost:7545'));
  });
};

const initContract = () => {
  const deploymentKey = Object.keys(Cert.networks)[0];
```

```
  return new web3.eth.Contract(Cert.abi, Cert.networks[deploymentKey].address);
};
let error = 0;
let college = sessionStorage.getItem('currentLoggedInCollege');
console.log(college);
let allColleges = JSON.parse(localStorage.getItem('collegesData'));
console.log(allColleges);
let col = allColleges.find(function (col) {
  return col.username === college;
});
let students = col.students || [];
const enrollForm = document.getElementById('enroll-form');
const studentID = document.getElementById('enroll-id');
const studentName = document.getElementById('enroll-sname');
const courseName = document.getElementById('enroll-cname');
const joiningDate = document.getElementById('enroll-date');

function setErrorFor(input, message) {
  const inputBox = input.parentElement;
  const small = inputBox.querySelector('small');
  small.innerText = message;
  error++;
  inputBox.className = 'inputBox error';
}

function setSuccessFor(input) {
  const inputBox = input.parentElement;
  inputBox.className = 'inputBox success';
}
const courses = ['B.Tech', 'M.Tech', 'MBA'];

function checkInputs() {
```

```
error = 0;
const studentIDValue = studentID.value.trim();
const studentNameValue = studentName.value.trim();
const courseNameValue = courseName.value.trim();
const joiningDateValue = joiningDate.value.trim();

//check ID
if (studentIDValue === '') {
  setErrorFor(studentID, 'Student ID cannot be empty');
} else {
  setSuccessFor(studentID);
}
//check name
if (studentNameValue === '') {
  setErrorFor(studentName, 'Student Name cannot be empty');
} else {
  setSuccessFor(studentName);
}
//check course
if (courseNameValue === '') {
  setErrorFor(courseName, 'Course Name cannot be empty');
} else if (!courses.includes(courseNameValue)) {
  setErrorFor(courseName, 'Enter a valid course name');
} else {
  setSuccessFor(courseName);
}
//check date
if (joiningDateValue === '') {
  setErrorFor(joiningDate, 'Joining date cannot be empty');
} else {
  setSuccessFor(joiningDate);
}
```

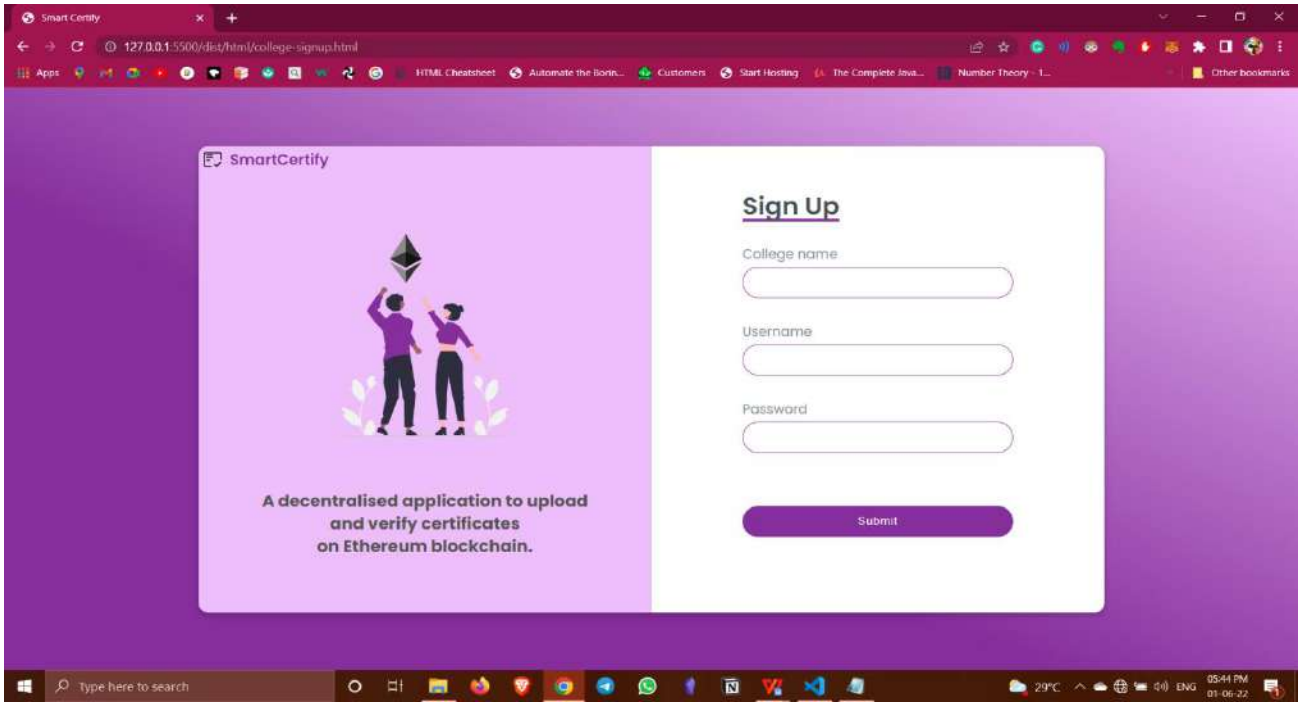
```
  return error;
}
const initApp = () => {
  let accounts = [];
  web3.eth.getAccounts().then(_accounts => {
    console.log(_accounts);
    accounts = _accounts;
  });
  enrollForm.addEventListener('submit', e => {
    e.preventDefault();
    const numErrors = checkInputs();
    if (numErrors === 0) {
      console.log('Success');
      let s = Object.create({});
      s.id = studentID.value.trim();
      s.name = studentName.value.trim();
      s.course = courseName.value.trim();
      s.joiningDate = joiningDate.value.trim();
      students.push(s);
      let college = sessionStorage.getItem('currentLoggedInCollege');
      console.log(college);
      // console.log(typeof localStorage.getItem('collegesData'));
      let allColleges = JSON.parse(localStorage.getItem('collegesData'));
      console.log(allColleges);
      let col = allColleges.find(function (col) {
        return col.username === college;
      });
      const index = allColleges.indexOf(col);
      col.students = students;
      allColleges[index] = col;
      cert.methods
        .enrollStudent(
```

```
        studentID.value,
        studentName.value,
        courseName.value,
        joiningDate.value
      )
      .send({ from: accounts[0] })
      .then(e => {
        console.log(e);
        console.log(
          `Student ${studentName.value} with id ${studentID.value} enrolled successfully`
        );
      })
      .catch(e => {
        console.log(e);
        console.log('Problem enrolling!!!');
      });
  }
 });
};
// console.log(localStorage.getItem('collegesData');
document.addEventListener('DOMContentLoaded', () => {
 initWeb3()
  .then(_web3 => {
   web3 = _web3;
   cert = initContract();
   initApp();
  })
  .catch(e => console.log(e.message));
});
```

5. SCREENSHOTS

5.1 SIGN UP PAGE

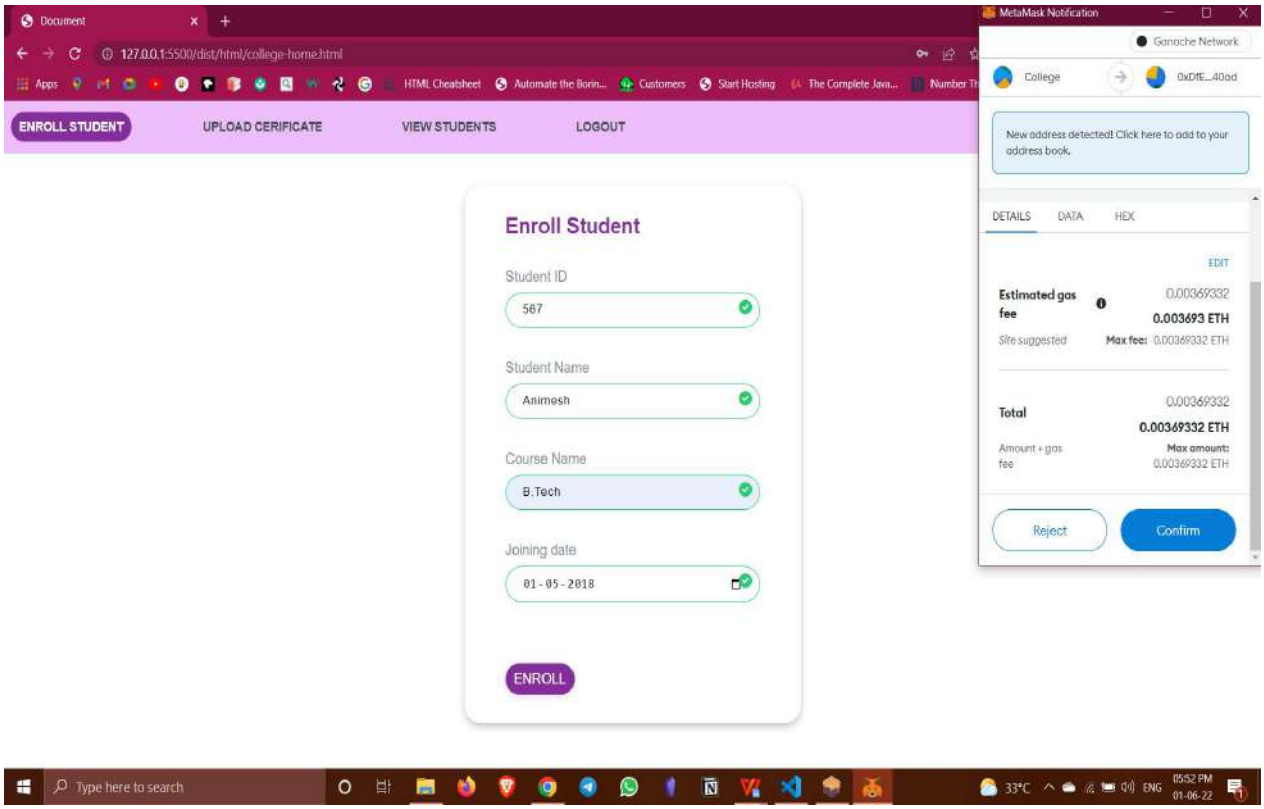
The sign page is where the user registers when they first visit the website. As shown in the screenshot 5.1, user should provide an email address and set a password. Respective account should provide a name, username and password.



Screenshot 5.1 : Sign up page

5.2 ENROLL STUDENTS

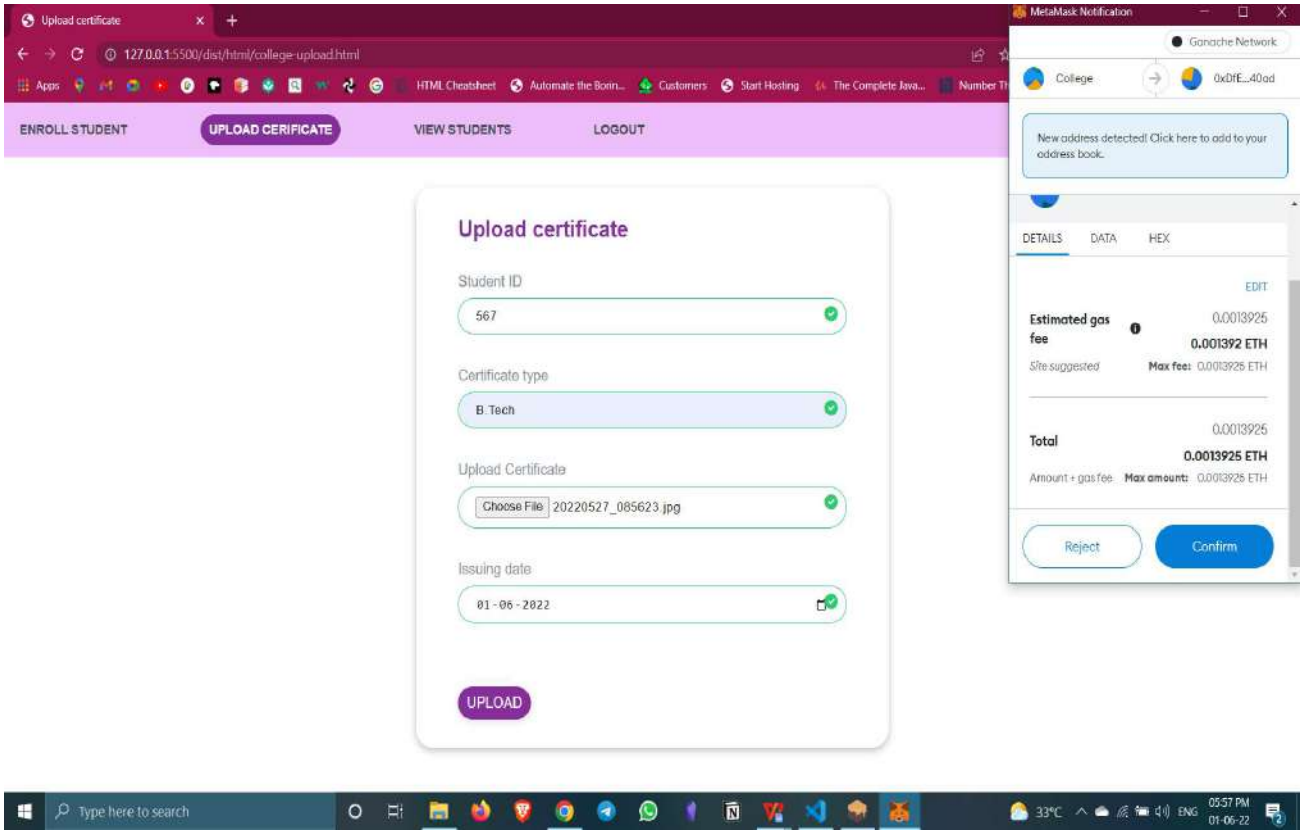
As shown in the screenshot 5.2, the enroll students page contains a form for enrolling the student. After all details are checked, then when clicked on enroll button a metamask notification is popped up for confirming the transaction. The gas fees will be deducted from the college's ethereum account after confirming the transaction and the student will be added to that college's database.



Screenshot 5.2 : Enroll Students

5.3 UPLOADING CERTIFICATE

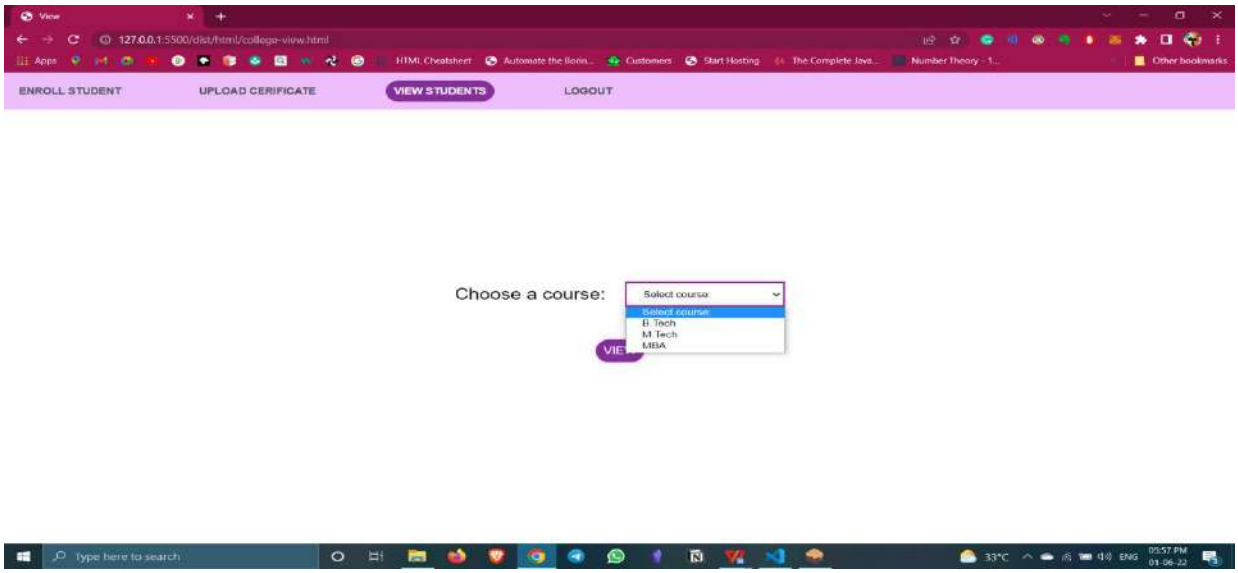
As shown in the screenshot 5.3, this page is used to upload student's credentials. Student ID, course type and the image of the certificate that has to be attached is provided during enrollment. Also, the joining date should also be provided. On clicking the upload button, metamask notification pops up to confirm the transaction being made.



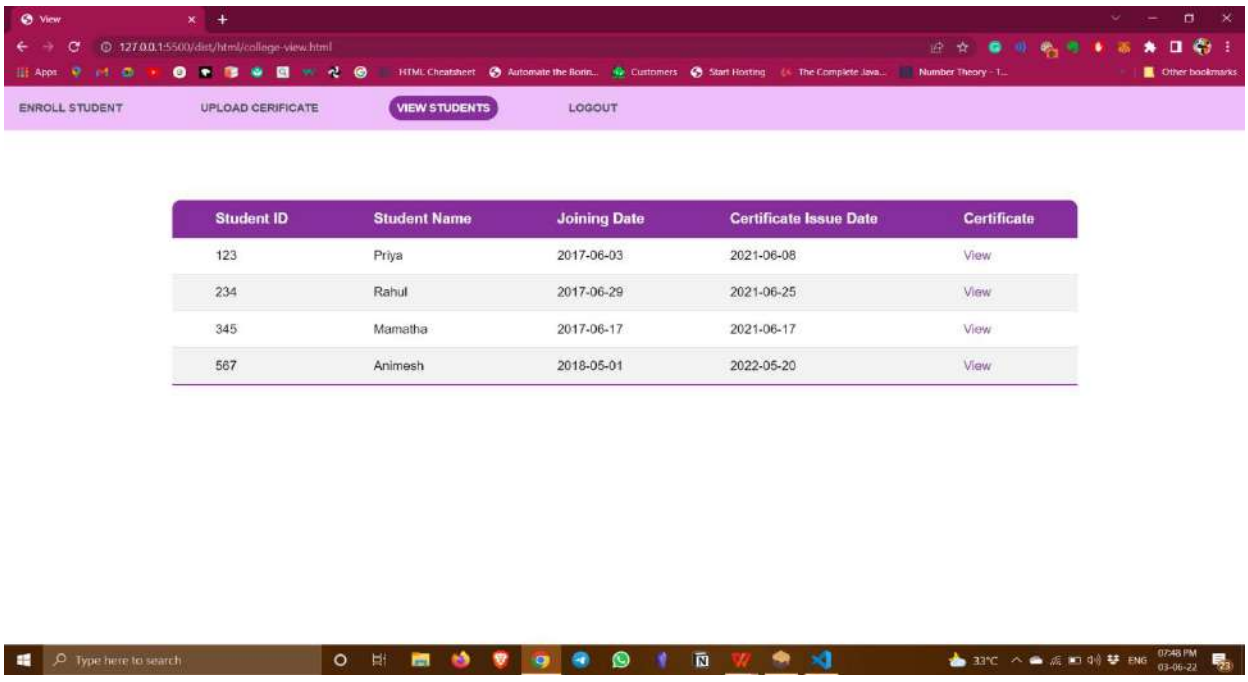
Screenshot 5.3 : Upload certificate

5.4 VIEW STUDENTS

As shown in screenshot 5.4, this page is to view student's information and also their certificate. For easier access, first we can choose a course type and click on the view button to view only students enrolled in that particular course.



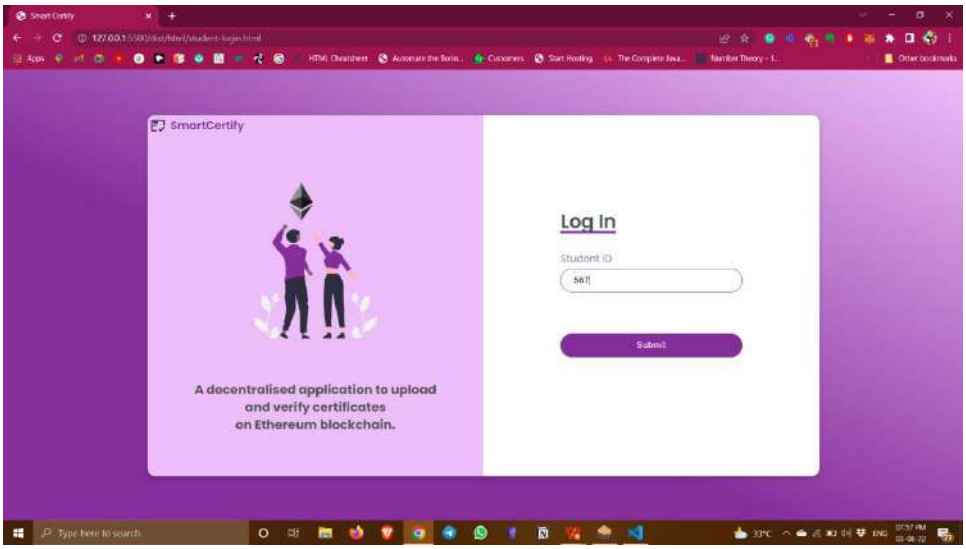
Screenshot 5.4 : View Students



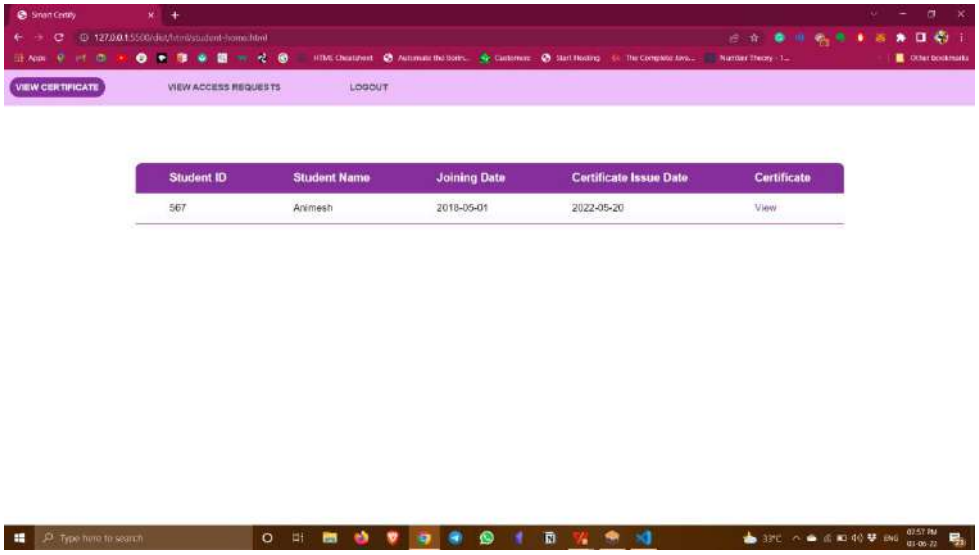
Screenshot 5.5: View Student Details

5.5 VIEW CERTIFICATE

As shown in screenshot 5.6, this is students login page and student can login using the enrollment ID provided to them by college. After logging in, student can view their details and credentials uploaded by college as shown in screenshot 5.7.



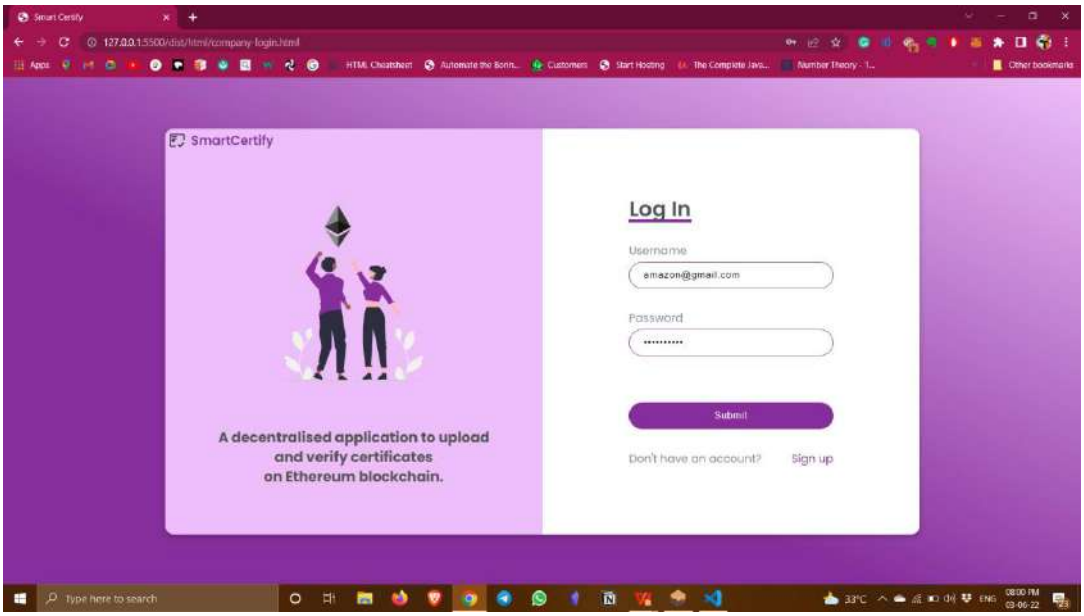
Screenshot 5.6: Students login page



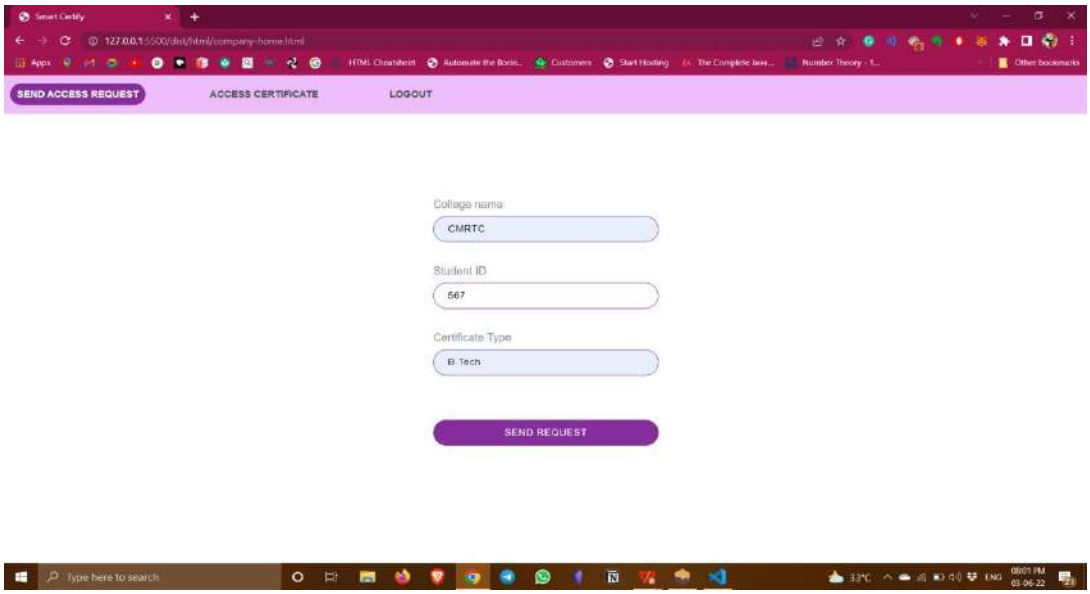
Screenshot 5.7: View certificate

5.6 SEND ACCESS REQUESTS

As shown in screenshot 5.8, company can login using their username and password and the company home page has two pages to navigate, one is send requests page as shown in screenshot 5.9 and other is to view the credentials of students who accepted the requests.



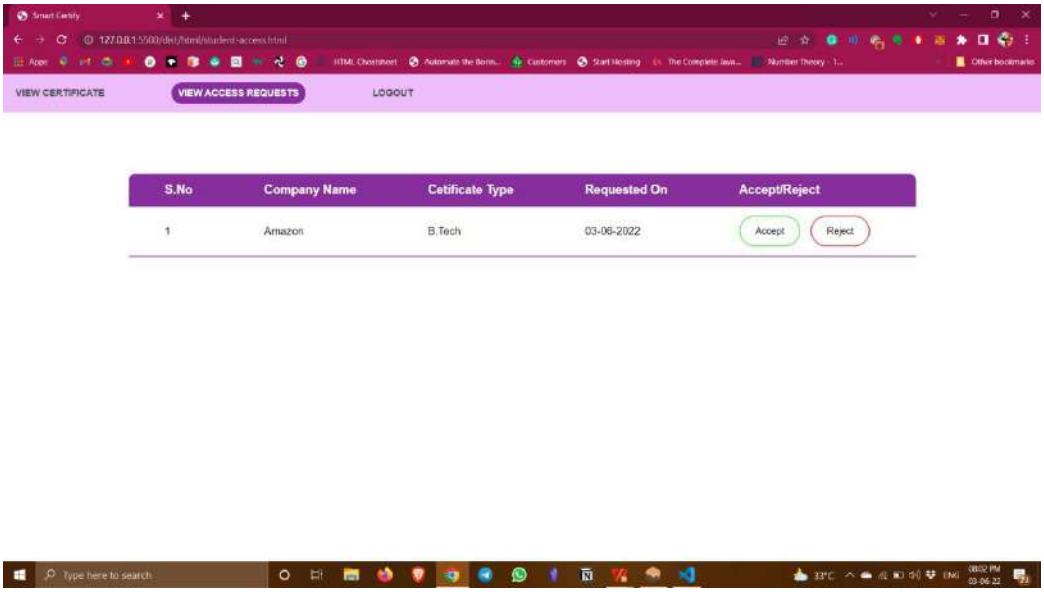
Screenshot 5.8: Company login page



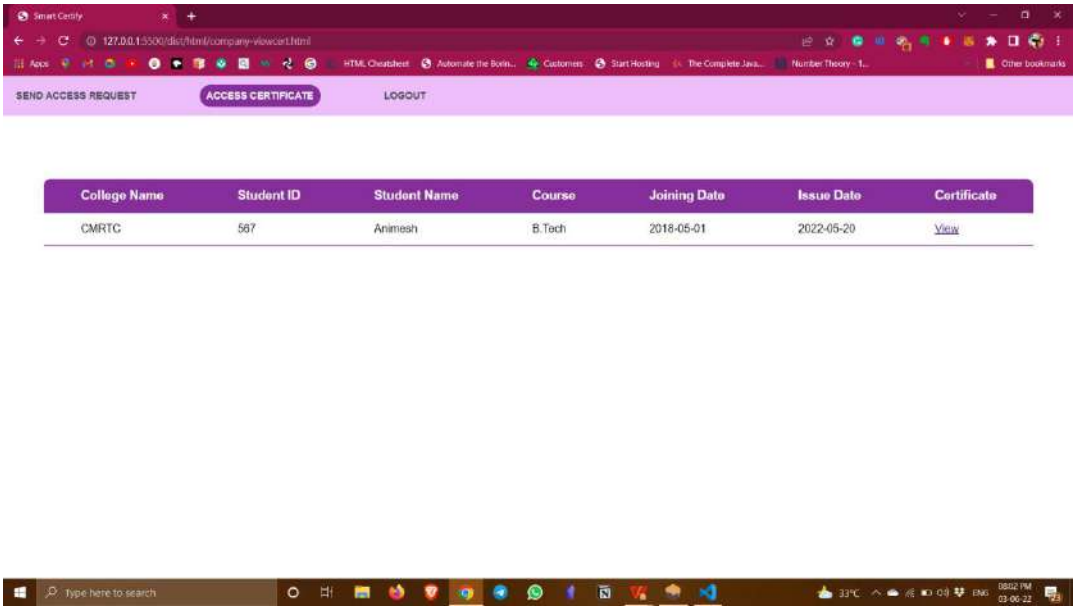
Screenshot 5.9: Send Access Request

5.7 ACCEPT ACCESS REQUESTS

As shown in screenshot 5.10, this is the students profile and on accept access requests page, all the requests send by different companies to this students are listed with an option of either rejecting the requests or accepting them. Once the request is accepted, the student credentials can be accessed by the companies on access certificates page on their profile as shown in screenshot 5.11.



Screenshot 5.10: View Access Requests



Screenshot 5.11: Access certificate

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discovery every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.3 TEST CASES

6.3.1 LOGIN

As shown in the table 6.1, the test cases for login functionality are taken. For the first test case, we check the functionality for when user enters correct username and password which makes the user to be successfully logged in and in the second test case we take the case where the user enters wrong credentials.

Test case ID	Test case name	Purpose	Test case	Output
1	User attempts to login	User authentication	User enters correct username and password	User logins successfully

2	User attempts to login	User authentication	User enter incorrect username or password	User login unsuccessful.

Table 6.1: Test cases for login

6.3.2 ENROLLING STUDENT

As shown in table 6.2, the test cases for enrolling student are checked. The first test case is to check if the student id is already existing. The second test case is to check if the other details like name, joining date and course type are valid.

Test case ID	**Test case name**	**Purpose**	**Test case**	**Output**
1	Student ID already exists	Testing enroll functionality	Student with same ID already exists in the database	Student enrollment unsuccessful.
2	Enters valid course type and joining date.	Testing enroll functionality	Enters valid details	Student enrolled successfully and transaction is recorded.

Table 6.2: Test cases for enrolling students

6.3.3 UPLOADING CERTIFICATE

The test cases for the main functionality, that is, uploading certificates on to the ethereum blockchain is checked as shown in table 6.3. A certificate cannot be uploaded more than once for a particular course of a particular student. The transaction is rejected in that case.

Test case ID	Test case name	Purpose	Test case	Output
1	Uploading certificate more than once.	Uploading certificate	College uploads certificate again for the same student	Transaction is rejected.
2	All data entered is valid	Uploading certificate	All data entered is correct.	Transaction is successful and certificate is uploaded on IPFS.

Table 6.3: Test cases for uploading certificate

7. CONCLUSION AND FUTURE SCOPE

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

The proposed system is a consortium blockchain among universities, their affiliated colleges, autonomous colleges, and the companies. Typically, universities first add the students' certificates and subsequently the companies can verify the credentials. The data stored in a blockchain will be protected as no one can tamper it or add new transactions to it with a back date. The generated unique ID for each transaction is later used to verify the certificates. This system can be used by all the universities and colleges, in order to provide extra security to the certificates and the students' data. The problem of fake certificates can be eradicated and there will be no question of its validation.

7.2 FUTURE SCOPE

In the future, this can be extended to provide integrity to any type of documents not only to the education sector but also to government sectors where a digital document time stamp is required. Not only to store the student marks information but also to store their employment and experience data, and can also be tracked by using this proposed system.

8. BIBILOGRAPHY

8. BIBILOGRAPHY

8.1 REFERENCES

[1] T.Rama Reddy, Rayudu Srinivas, "Proposing a reliable method of securing and verifying the credentials of graduates through blockchain", Published on Springer in June 2021.

[2] Nikhil Gaikwad, Nevil D'Souza, "A Blockchain-based verification system for academic certificates", Published on IEEE in September 2021.

[3] Nero Chaniago, Parman Sukamo, "Electronic document authenticity verification of diploma and transcript using smart contract on Ethereum blockchain", Published on ResearchGate in May 2021.

8.2 WEBSITES

[1] https://web3js.readthedocs.io/en/v1.7.3/

[2] https://developer.mozilla.org/en-US/docs/Web/JavaScript

[3] https://trufflesuite.com/docs/

[4] https://docs.ipfs.io/

[5] https://docs.soliditylang.org/en/v0.8.14/

8.3 LINKS

GitHub - git@github.com:Charitha2588/Major-project.git