VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Project Report

on

DECENTRALIZED UNIVERSITY FEES TRANSACTION

to be submitted in partial fulfilling of the requirements for the course on

Network and Information Security – ITA6007

VL2022230500484 (D1+TD1)

by

SHIVAM SINGH - 22MCA0257

Faculty Name: Dr C. NAVANEETHAN

(ASSOCIATE PROFESSOR SENIOR - SITE)

Winter Semester 2022-2023

TABLE OF CONTENTS

TITLE **PAGE NO.**

ABSTRACT

In recent years, the rise of blockchain technology has revolutionized various industries, including finance, supply chain management, and healthcare. This research paper focuses on the application of blockchain and the Digital Signature Algorithm (DSA) in the context of decentralized university fee transactions.

Traditional methods of processing university fees involve multiple intermediaries, leading to inefficiencies, delays, and increased costs. By leveraging blockchain technology, this research proposes a decentralized approach to university fee transactions, eliminating the need for intermediaries and providing a transparent and secure platform for students, universities, and other stakeholders.

The Digital Signature Algorithm (DSA) is a widely adopted cryptographic algorithm that provides authentication and integrity to digital transactions. This paper explores the integration of DSA within the decentralized university fee transaction system to ensure the authenticity of transactions and prevent fraud.

The proposed system utilizes smart contracts deployed on a blockchain network, enabling automatic execution of fee transactions once predefined conditions are met. The blockchain's distributed ledger records all transactions, ensuring transparency and auditability.

Through a comprehensive evaluation and simulation of the system, this research demonstrates the feasibility and benefits of implementing decentralized university fee transactions using the Digital Signature Algorithm. The results indicate improved efficiency, reduced costs, enhanced security, and a more streamlined experience for all participants.

Keyword: Blockchain, Decentralized transactions, university fees, Digital Signature Algorithm, Smart contracts

INTRODUCTION

With the advent of blockchain technology, various sectors have witnessed significant transformations in the way transactions are conducted. The education sector, particularly in the context of university fee transactions, can greatly benefit from decentralized and secure systems. This research aims to explore the implementation of a decentralized university fee transaction system using the Digital Signature Algorithm (DSA).

Traditional methods of processing university fees involve multiple intermediaries, such as banks and payment processors, resulting in cumbersome processes, delays, and increased costs. Moreover, these centralized systems are prone to security vulnerabilities and lack transparency, leading to concerns regarding the integrity of fee transactions. To address these issues, a decentralized approach utilizing blockchain technology is proposed.

The motivation behind this project stems from the need to streamline and enhance the efficiency of university fee transactions. By leveraging the inherent characteristics of blockchain, including immutability, transparency, and security, this research aims to provide a more reliable and seamless experience for students, universities, and other stakeholders involved in the fee payment process.

The existing system faces several challenges. Intermediaries introduce complexities and delays in the payment process, leading to inconvenience for students and administrative inefficiencies for universities. Furthermore, security risks, such as unauthorized access and fraudulent transactions, pose significant threats to the integrity of fee payments.

The contributions of this proposed work are twofold. Firstly, it leverages the Digital Signature Algorithm (DSA), a widely adopted cryptographic algorithm, to ensure the authenticity and integrity of fee transactions. DSA provides strong authentication, preventing unauthorized access and tampering of transaction data. Secondly, by implementing a decentralized system using blockchain technology, this research aims to eliminate intermediaries, enhance transparency, and provide a more efficient platform for university fee transactions.

Through a comprehensive evaluation and simulation, this research will demonstrate the feasibility and benefits of the proposed decentralized university fee transaction system. The expected outcomes include improved efficiency, reduced costs, enhanced security, and a seamless experience for all participants involved in the fee payment process.

LITERATURE REVIEW

NO.	TITLE	CHALLENGES	METHODOLOGY	APPLICATIONS
1	***"A secure blockchain-based payment system for university fees"***	Challenges: The paper identifies challenges such as the need for secure authentication, scalability, and privacy in university fee transactions.	Methodology: The paper proposes a blockchain-based payment system that utilizes smart contracts and cryptographic techniques to ensure secure and efficient fee transactions.	Applications: The system is designed specifically for university fee payments, providing a decentralized and transparent platform for students and universities.
2	***"An approach to decentralized payment systems using blockchain technology"***	Challenges: The paper discusses challenges related to the existing centralized payment systems, including the reliance on intermediaries and lack of transparency.	Methodology: The paper presents a decentralized payment system using blockchain technology, allowing direct peer-to-peer transactions without intermediaries.	Applications: The proposed system can be applied to various industries, including university fee transactions, to eliminate intermediaries and enhance transparency.
3	***"Blockchain technology for secure and privacy-preserving e-health systems"***	Challenges: The paper highlights challenges in e-health systems, such as data privacy and security concerns.	Methodology: The paper explores the use of blockchain technology to create a secure and privacy-preserving e-health system, ensuring data integrity and patient privacy.	Applications: The proposed system can be applied to securely manage electronic health records, ensuring privacy and preventing unauthorized access.
4	***"Blockchain technology in education: The potential of a decentralized student record"***	Challenges: The paper addresses challenges related to the centralized management of student records, including data security and authentication.	Methodology: The paper discusses the potential of blockchain technology to create a decentralized student record management system, ensuring data security, authentication, and ease of verification.	Applications: The proposed system can be applied to securely store and share student records, simplifying the verification process for educational institutions.
5	***"Blockchain and research funding: A proposal for an innovative and transparent approach"***	Challenges: The paper highlights challenges in the existing research funding systems, such as lack of transparency and accountability.	Methodology: The paper proposes a blockchain-based system for research funding, ensuring transparency, accountability, and traceability of funds.	Applications: The system can be applied to enhance transparency and efficiency in the allocation and tracking of research funding.
6	***"Blockchain-Based Credentialing for Higher Education: A Comparison of the Bitcoin and Ethereum Networks"***	Challenges: The paper discusses challenges related to credentialing in higher education, including the need for secure and tamper-proof credentials.	Methodology: The paper compares the Bitcoin and Ethereum networks as potential platforms for blockchain-based credentialing systems in higher education.	Applications: The proposed system can be applied to securely issue and verify academic credentials, preventing fraud and simplifying the verification process.
7	***"Blockchain-Based Payment System for Online Education"***	Challenges: The paper identifies challenges in the existing payment systems for online education, such as high transaction fees and payment delays.	Methodology: The paper proposes a blockchain-based payment system for online education, utilizing smart contracts to facilitate secure and efficient fee transactions.	Applications: The system is designed specifically for online education platforms, providing a decentralized and cost-effective payment solution.
8	***"A blockchain-based identity management system for students"***	Challenges: The paper addresses challenges in traditional identity management systems, including	Methodology: The paper proposes a blockchain-based identity management system for students,	Applications: The proposed system can be applied to student identification and

		the risk of identity theft and lack of control over personal data.	enabling individuals to have control over their personal data and securely authenticate their identities.	authentication processes, enhancing security and privacy.
9	***"A blockchain based voting system for universities"***	Challenges: The paper discusses challenges in the existing voting systems in universities, such as ballot manipulation and lack of transparency.	Methodology: The paper proposes a blockchain-based voting system for universities, ensuring transparency, immutability, and secure tallying of votes.	Applications: The system can be applied to conduct fair and transparent elections in universities, preventing fraud and ensuring the integrity of the voting process.
10	***"Blockchain-based student record management system"***	Challenges: The paper addresses challenges related to the centralized management of student records, including data security and authentication.	Methodology: The paper proposes a blockchain-based student record management system, providing secure storage, tamper-proof records, and ease of verification.	Applications: The system can be applied to securely manage and share student records, simplifying the verification process for educational institutions.
11	***"An efficient blockchain-based framework for university fee payments"***	Challenges: The paper identifies challenges in the existing university fee payment systems, including delays, high transaction costs, and lack of transparency.	Methodology: The paper presents an efficient blockchain-based framework for university fee payments, utilizing smart contracts and cryptographic techniques to streamline the payment process.	Applications: The proposed framework can be applied to universities' fee payment systems, reducing transaction costs, improving efficiency, and ensuring transparency.
12	***"An approach to decentralized payment systems using blockchain technology"***	Challenges: The paper discusses challenges related to the existing centralized payment systems, including the reliance on intermediaries and lack of transparency.	Methodology: The paper presents an approach to decentralized payment systems using blockchain technology, enabling direct peer-to-peer transactions without intermediaries.	Applications: The proposed approach can be applied to various industries, including university fee transactions, to eliminate intermediaries and enhance transparency.
13	***"An efficient blockchain-based framework for university fee payments"***	Challenges: The paper identifies challenges in the existing university fee payment systems, including delays, high transaction costs, and lack of transparency.	Methodology: The paper proposes an efficient blockchain-based framework for university fee payments, utilizing smart contracts and cryptographic techniques to streamline the payment process.	Applications: The proposed framework can be applied to university fee payment systems, reducing transaction costs, improving efficiency, and ensuring transparency.

PROPOSED SYSTEM

Overview

The proposed system aims to create a decentralized university fee transaction system using the Digital Signature Algorithm (DSA) and the Ethereum blockchain. This system provides an efficient and secure platform for students to pay their course fees and ensures transparency in the transaction process. The system incorporates a web-based interface where users can log in and enroll in their desired courses. The system leverages the Ethereum blockchain for maintaining a secure and immutable ledger of fee transactions.

System Architecture

The system architecture consists of multiple components, including the web-based interface, the Ethereum blockchain, and the smart contract deployed on the blockchain. The web-based interface is developed using TypeScript and CSS, providing a user-friendly portal for students to interact with the system. The Ethereum blockchain serves as the underlying decentralized infrastructure, facilitating secure and transparent fee transactions. The smart contract, written in Solidity, acts as the business logic layer, executing the fee payment transactions and enforcing the necessary rules and validations.

FUNCTIONAL ARCHITECTURE

The functional architecture of the system involves several key modules, each serving a specific purpose in the fee transaction process. These modules include user authentication, course selection, transaction confirmation, and payment processing. Each module plays a crucial role in ensuring a smooth and secure user experience.

1. **User Authentication:** This module verifies the identity of the user logging into the system, ensuring that only authorized individuals can access the fee payment portal.

2. **Course Selection:** Once authenticated, users can browse through the available courses and select their desired course. The system provides a comprehensive list of courses, such as MCA, M.Sc Data Science, B.Tech CSE, M.Sc Physics, and more.

3. **Transaction Confirmation:** After selecting a course, users are presented with a confirmation prompt. They can review the course details and associated fees before confirming or rejecting the transaction. This step ensures that users have full control over their fee payment decisions.

4. **Payment Processing:** Once the user confirms the transaction, they are redirected to a payment page. The selected course and the corresponding fee amount are displayed, and users can initiate the payment by clicking the "Pay Fees" button. Upon confirmation, the system initiates the transaction using the Digital Signature Algorithm and executes the payment from the user's account.

MODULAR DESIGN

The proposed system follows a modular design approach to enhance maintainability and extensibility. Each module incorporates well-defined algorithms and flow structures to ensure smooth workflow and user interaction. For example, the user authentication module employs a secure login algorithm to validate user credentials, while the payment processing module utilizes the DSA algorithm for secure fee transactions. Diagrams are used to depict the sequential steps involved in each module, providing a clear understanding of our system's workflow.

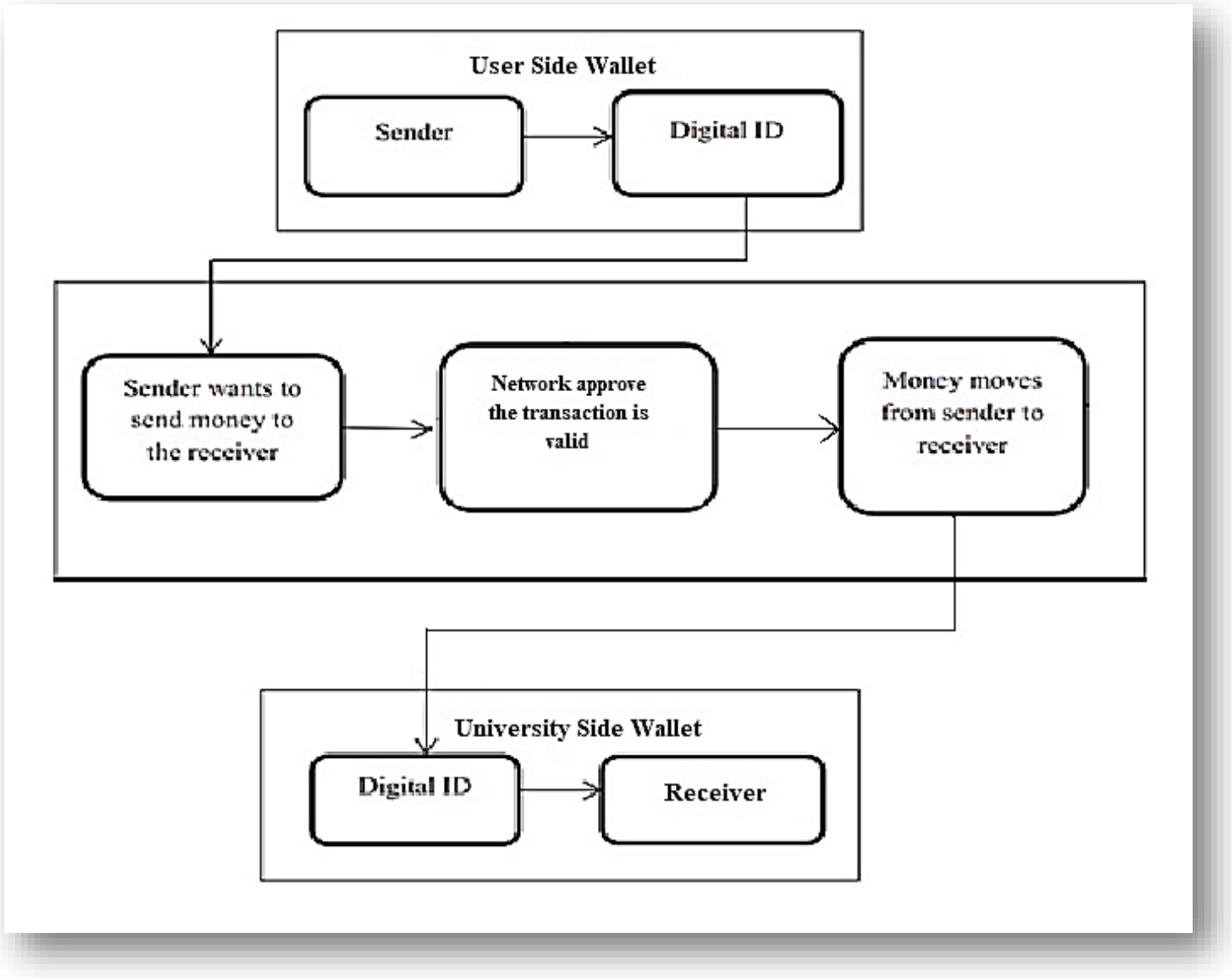


Figure: Architectural Design of the system

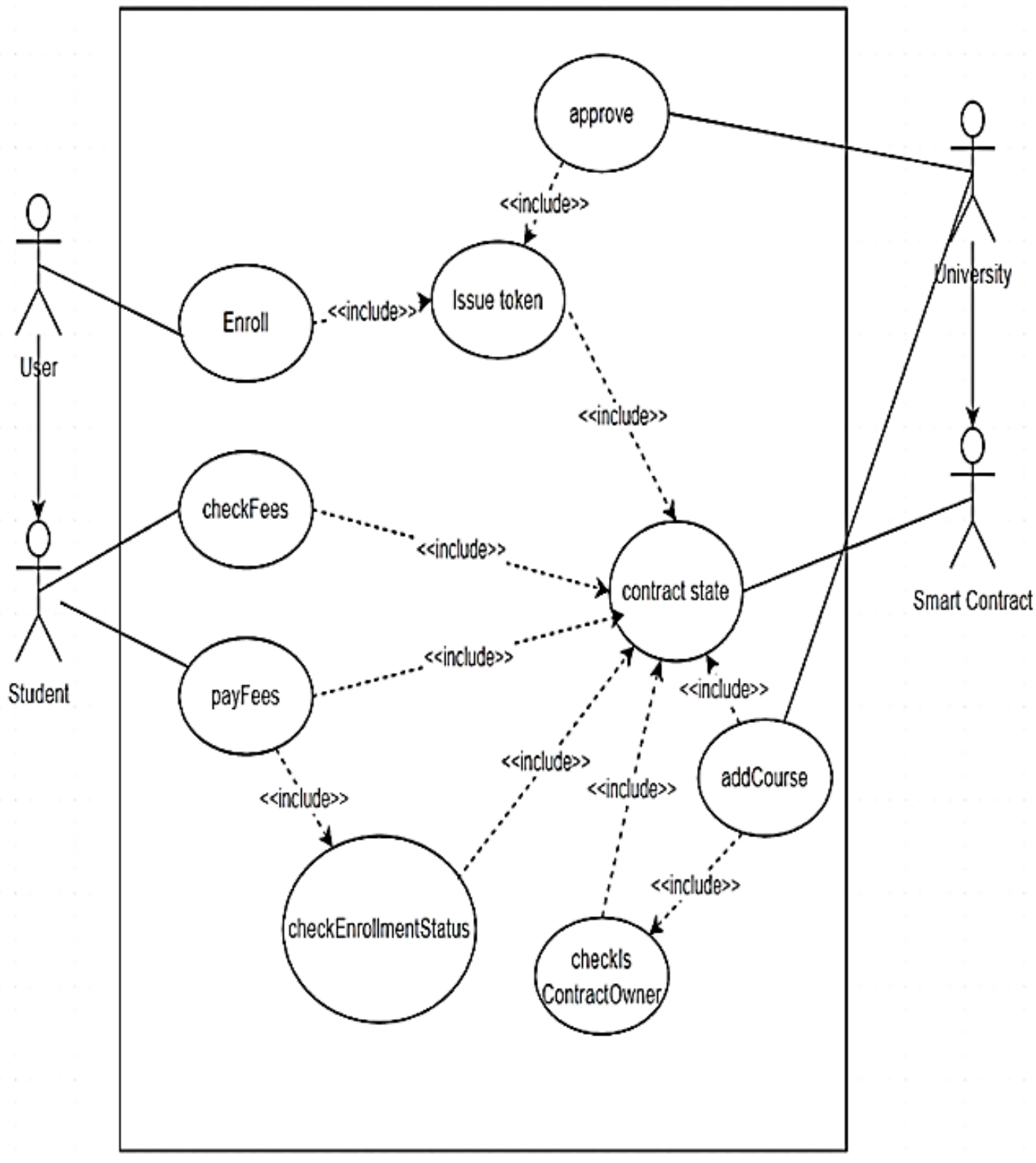


Figure: Use-Case Diagram for our system

INNOVATIVE IDEA IN OUR PROJECT

Compared to existing systems, our project incorporates several innovative features. First, the utilization of the Ethereum blockchain ensures a decentralized and transparent fee transaction process, eliminating the need for intermediaries and enhancing trust. The integration of the Digital Signature Algorithm adds an extra layer of security, ensuring the authenticity and integrity of fee transactions. Additionally, the web-based interface provides an intuitive and user-friendly platform for students to conveniently enroll in courses and make fee payments. The modular design approach ensures scalability and flexibility, allowing for easy integration of additional features in the future. By deploying a local blockchain network using Ganache, the system provides a testing environment for developers to simulate real-world scenarios and validate the system's functionality before deployment to the main Ethereum network.

Moreover, in the literature survey we found out that none of the paper, or journals has actually implemented the solution to this particular problem statement in a GUI based system. Most of them are conducting tests and comparative analysis on various system efficacy on blockchain. So, in that regard I consider this project unique, that it is providing the end user with a GUI through which they can interact in real-time to transact in a safe and secure manner on the blockchain network.

SOFTWARE DETAILS AND SCREENSHOTS

The implementation of the decentralized university fee transaction system involves several software components and user interfaces. Here are the software details and corresponding screenshots of the transaction web interface:

1. Solidity Smart Contract:

- The smart contract is written in Solidity, a programming language specifically designed for Ethereum smart contracts. It contains the necessary functions and variables to facilitate fee transactions and maintain the transaction records on the blockchain.

1

university.sol

contracts > university.sol

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

contract University {
    uint64 courseIdCounter; //! for keeping track of index for course.
    address contractOwner;
    address payable vaultAddress;

    struct Course {
        uint64 courseId;
        string courseName;
        uint256 fees;
    }

    struct Student {
        address walletAddress;
        uint64 enrolledCourseId;
        bool feesPaid;
    }

    Course[] universityCourses;
    Student[] enrolledStudents;

    constructor() {
        courseIdCounter = 0;
        contractOwner = msg.sender;
        vaultAddress = payable(contractOwner);
    }

    function getFees( string memory courseName ) public view returns( uint256 ) {
        int64 courseIndex = indexOfCourse(courseName);

        if ( courseIndex >= 0)
            return universityCourses[ uint64(courseIndex) ].fees;

        return 0;
    }
```

2

```
PASS src/index.spec.ts
  Deploying a contract
    ganache server
      ✓ is up (1 ms)
      ethers client
        ✓ can connect to ganache server (102 ms)
      accounts
        ✓ has data in it
      smart contract deployment
        owner 0x90f8bf6a479f320ead074411a4b0e7944ea8c9c1
          ✓ is owner of the smart contract (26 ms)
          ✓ can add courses to the contract (205 ms)
          ✓ returns all courses in the list (57 ms)
          ✓ contract matches the new amount of courses added (18 ms)
        random wallet -> 0x95ced938f7991cd0dfcb48f0a06a40fa1af46ebc
          ✓ can interact with the contract (14 ms)
          ✓ can request for course fees (21 ms)
          ✓ can query for course details (1 ms)
          ✓ can enroll as student (180 ms)
          ✓ enrolled for MCA (19 ms)
          ✓ can get his fees (20 ms)
          ✓ can pay fees (103 ms)

Test Suites: 1 passed, 1 total
Tests:       14 passed, 14 total
Snapshots:   0 total
Time:        2.667 s, estimated 3 s
Ran all test suites.
```

Figure 1, 2: *Solidity smart contract code for the decentralized university fee transaction system.*

2. TypeScript Frontend Code:

- The frontend code is developed using TypeScript, which enables interaction with the Ethereum blockchain and the smart contract. It includes functions for user authentication, course selection, transaction confirmation, and payment processing.

3

Decentralized-University-Transaction-main - newuser.tsx

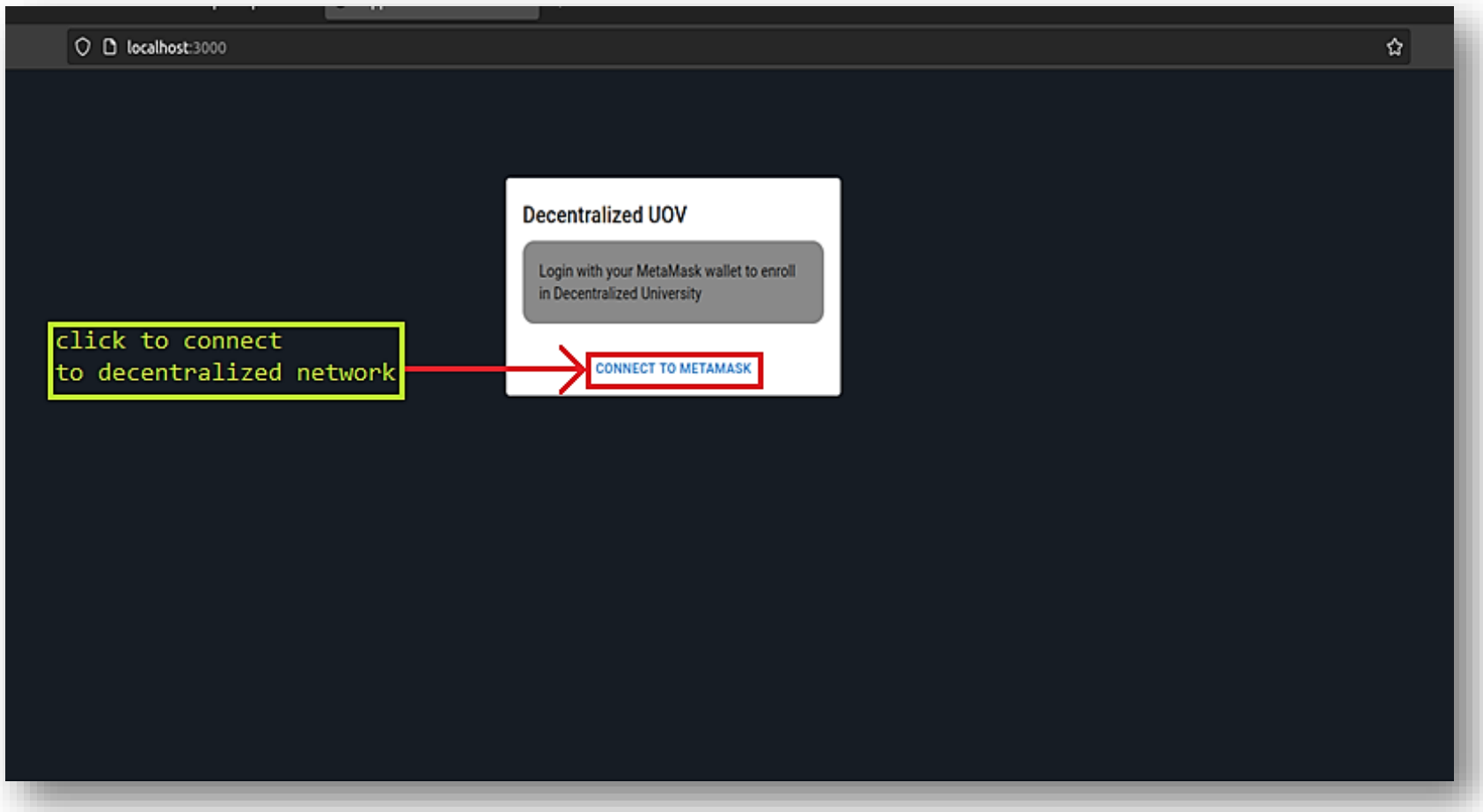
```tsx
import React from 'react'
import Router from 'next/router'
import Web3 from 'web3'
import { AbiItem } from 'web3-utils'
import { Contract } from 'web3-eth-contract'

import localContractDetails from '../../contract.json'
import ContractAbi from '../../../contracts/university_sol_University.json'

import { UserType } from '../../enums';

import { Button, Container } from '@mui/material'
import Table from '@mui/material/Table';
import TableBody from '@mui/material/TableBody';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TableRow from '@mui/material/TableRow';
import Paper from '@mui/material/Paper';

import NavigationBar from '../../components/navigation'

type State = {
    courses: Array<CourseDetail>,
    walletAddress: string,
    enrolledCourse: number
}

type Props = {

}
```

Figure 3: *TypeScript frontend code for the decentralized university fee transaction system.*

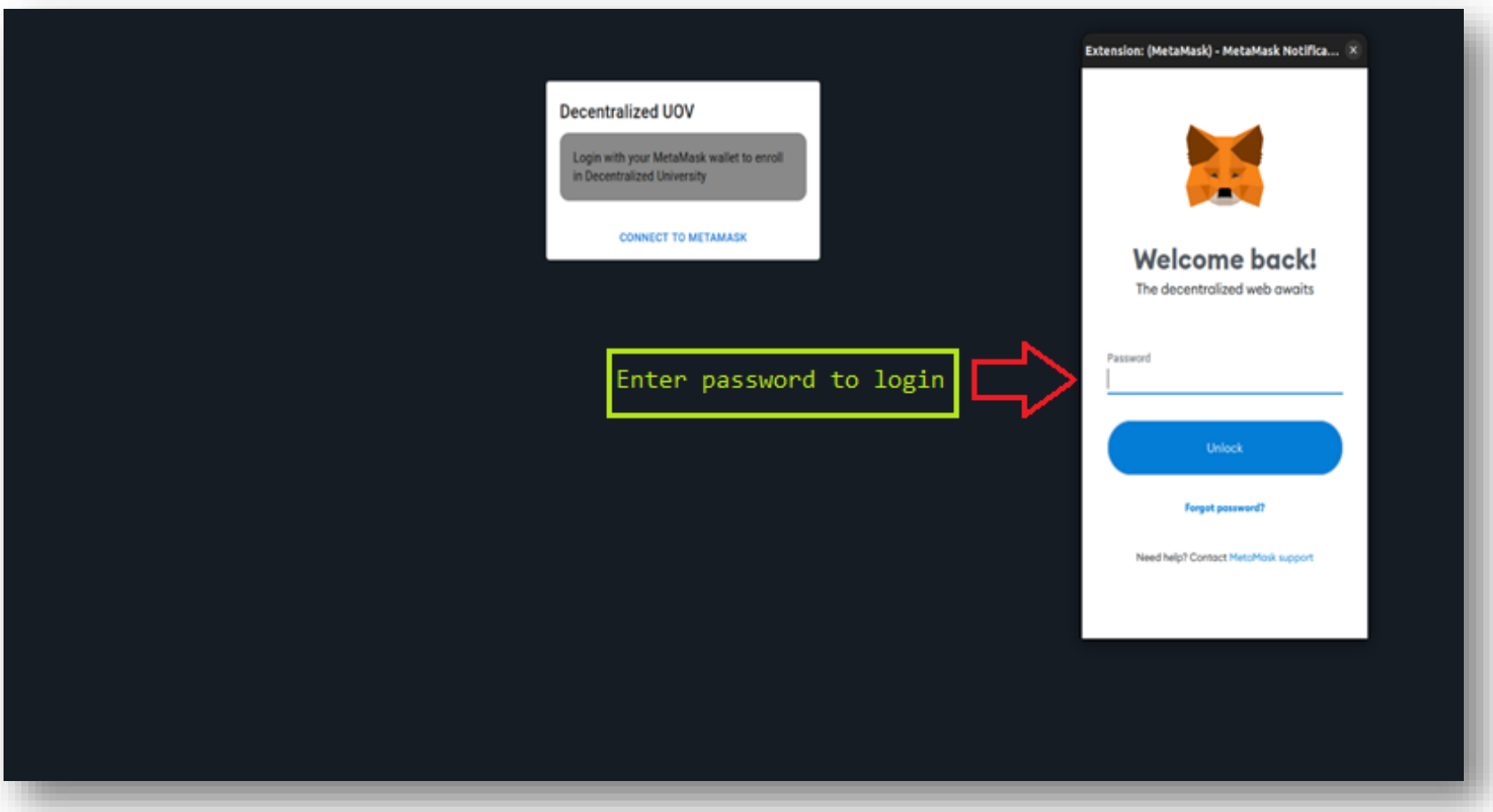
3. Web-Based Interface:

- The web-based interface provides a user-friendly platform for students to interact with the system. It includes login screens, course selection pages, transaction confirmation prompts, and payment processing screens.

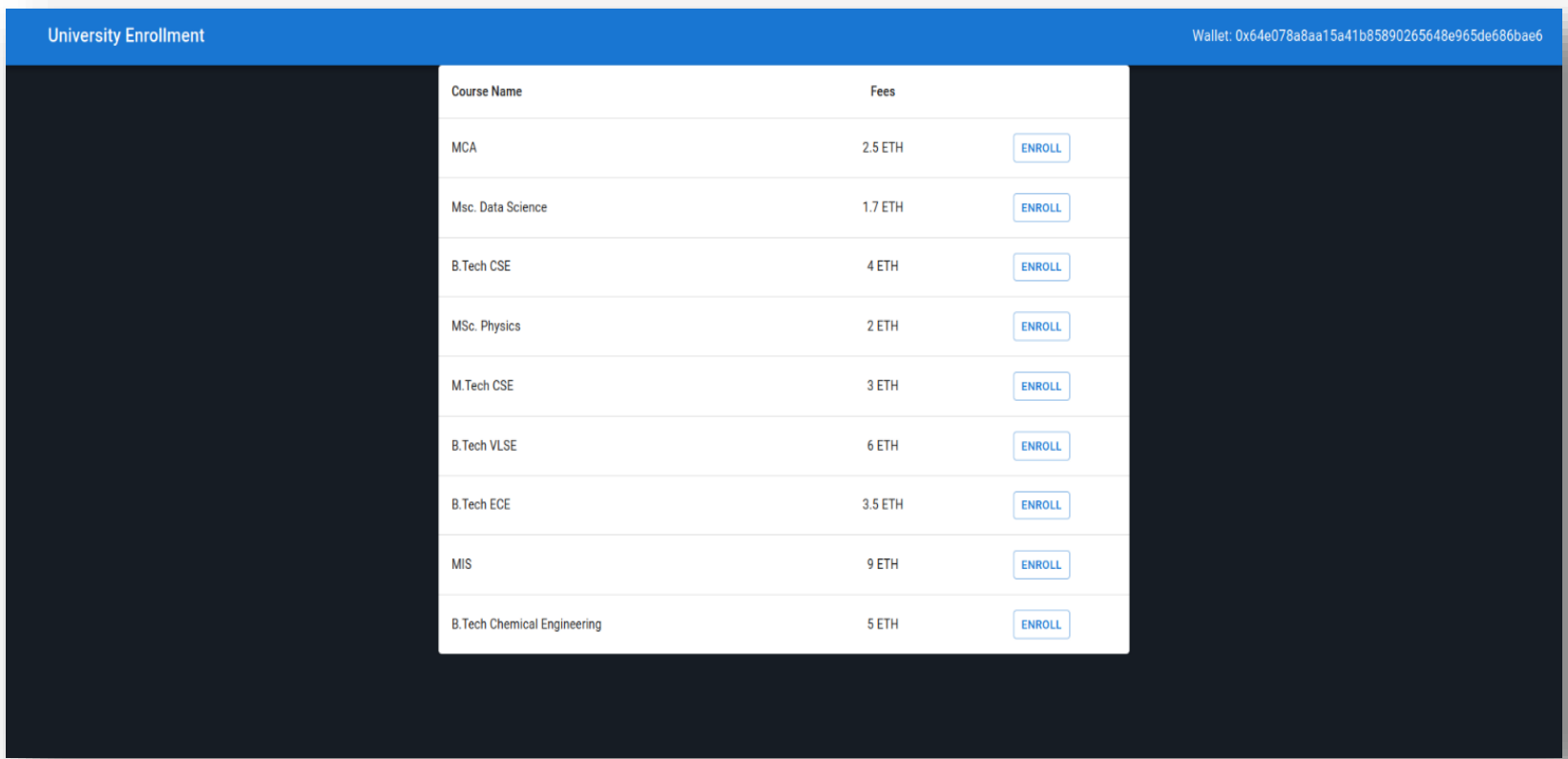
4



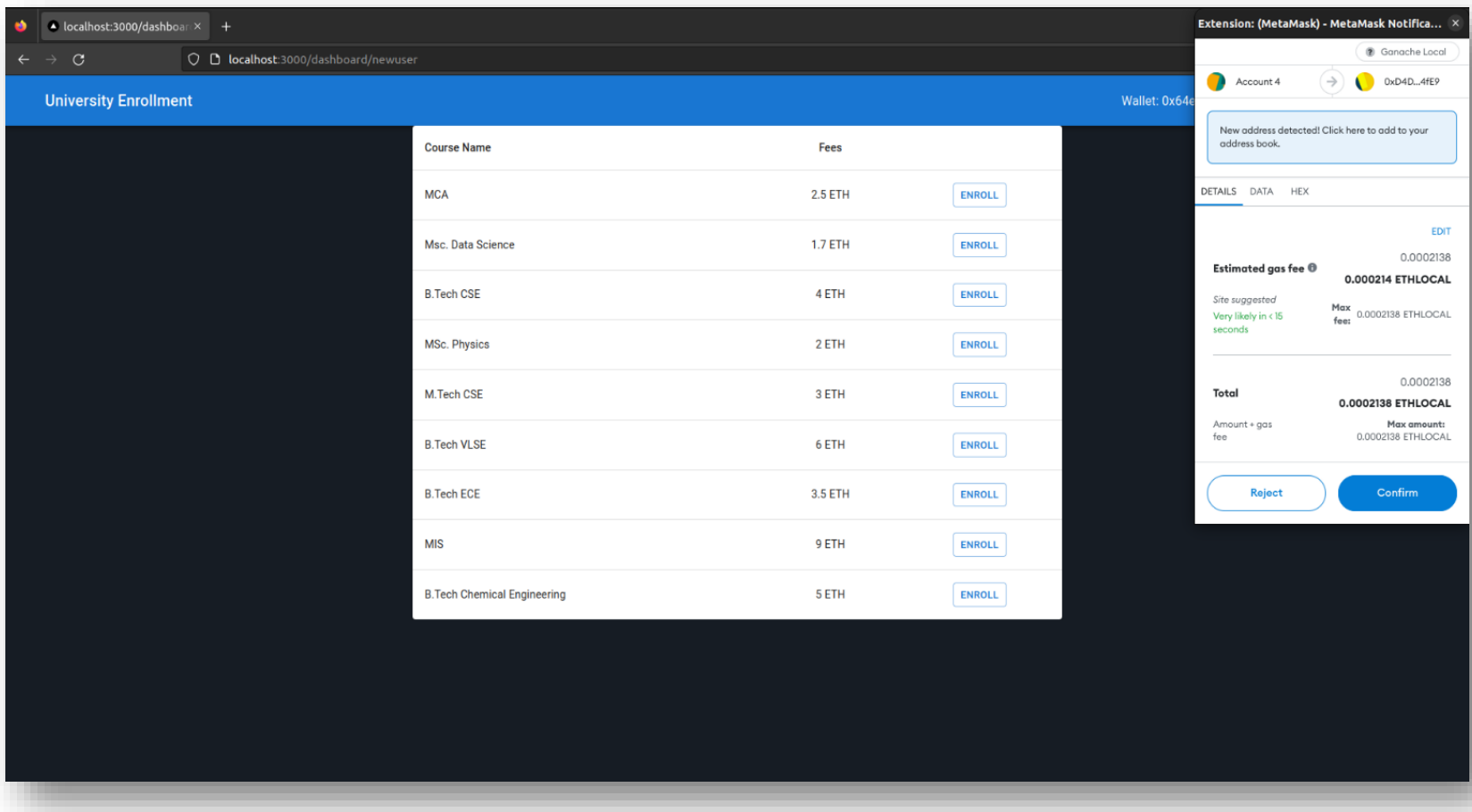
5



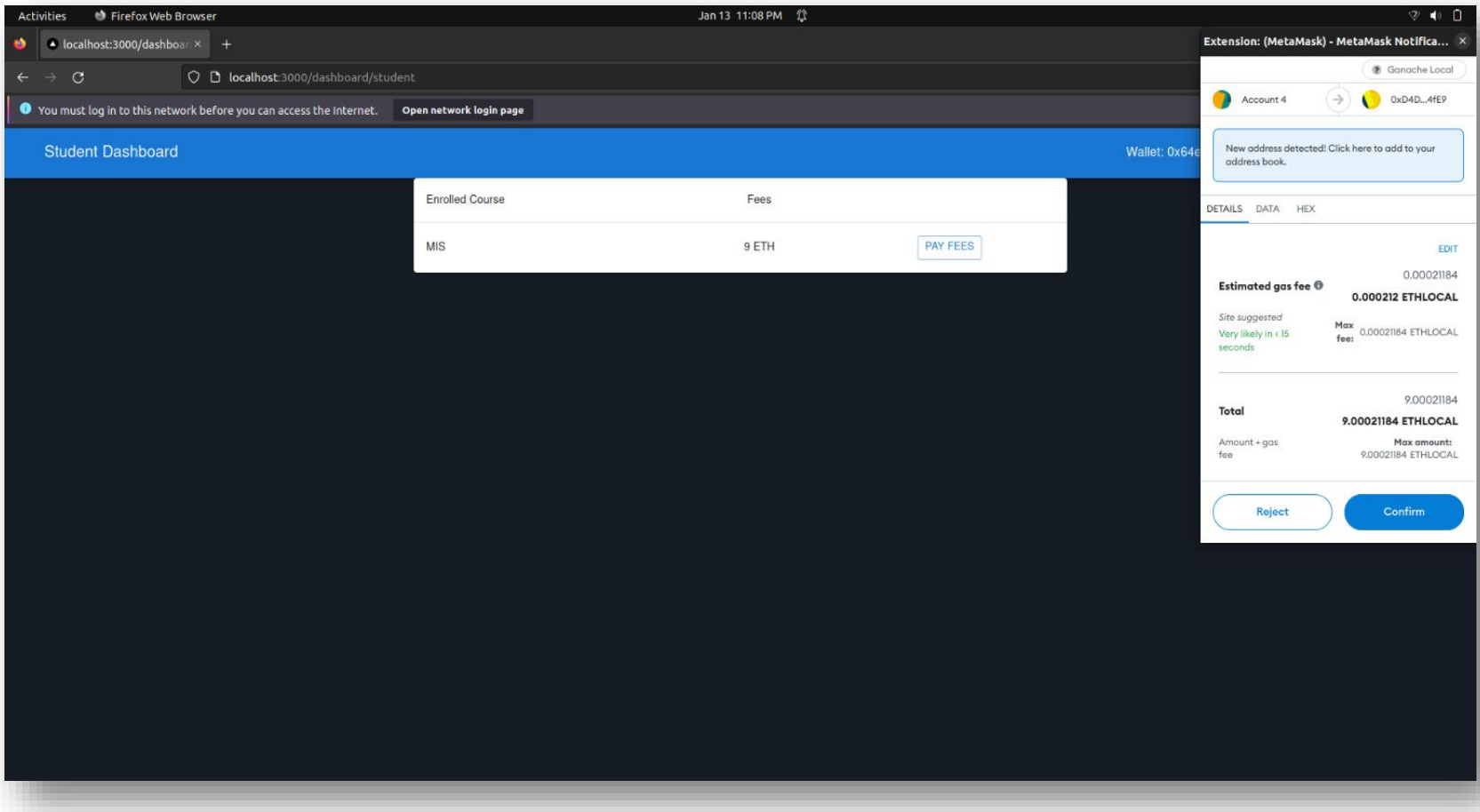
6


University Enrollment
Wallet: 0x64e078a8aa15a41b85890265648e965de686bae6
Course Name
Fees
MCA
2.5 ETH
ENROLL
Msc. Data Science
1.7 ETH
ENROLL
B.Tech CSE
4 ETH
ENROLL
MSc. Physics
2 ETH
ENROLL
M.Tech CSE
3 ETH
ENROLL
B.Tech VLSE
6 ETH
ENROLL
B.Tech ECE
3.5 ETH
ENROLL
MIS
9 ETH
ENROLL
B.Tech Chemical Engineering
5 ETH
ENROLL

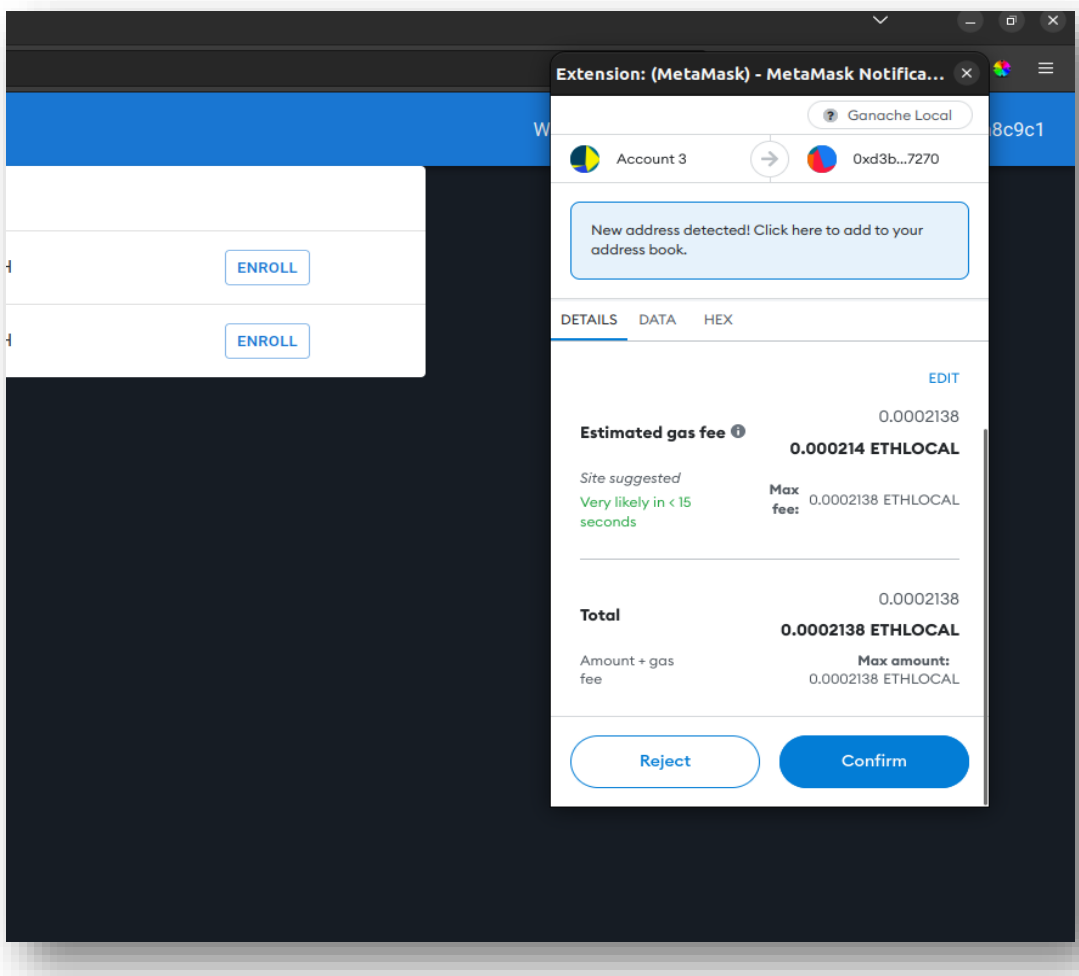
7


localhost:3000/dashboard/newuser
University Enrollment
Course Name
Fees
MCA
2.5 ETH
ENROLL
Msc. Data Science
1.7 ETH
ENROLL
B.Tech CSE
4 ETH
ENROLL
MSc. Physics
2 ETH
ENROLL
M.Tech CSE
3 ETH
ENROLL
B.Tech VLSE
6 ETH
ENROLL
B.Tech ECE
3.5 ETH
ENROLL
MIS
9 ETH
ENROLL
B.Tech Chemical Engineering
5 ETH
ENROLL
Extension: (MetaMask) - MetaMask Notifica...
Ganache Local
Account 4
0xD4D...4fE9
New address detected! Click here to add to your address book.
DETAILS
DATA
HEX
EDIT
Estimated gas fee
0.0002138
0.000214 ETHLOCAL
Site suggested
Very likely in < 15 seconds
Max fee:
0.0002138 ETHLOCAL
Total
0.0002138
0.0002138 ETHLOCAL
Amount + gas fee
Max amount:
0.0002138 ETHLOCAL
Reject
Confirm

8


Activities
Firefox Web Browser
Jan 13 11:08 PM
localhost:3000/dashboar
localhost:3000/dashboard/student
You must log in to this network before you can access the Internet.
Open network login page
Student Dashboard
Wallet: 0x64e
Enrolled Course
Fees
MIS
9 ETH
PAY FEES
Extension: (MetaMask) - MetaMask Notifica...
Ganache Local
Account 4
0xD4D...4fE9
New address detected! Click here to add to your address book.
DETAILS
DATA
HEX
EDIT
Estimated gas fee
0.00021184
0.000212 ETHLOCAL
Site suggested
Very likely in < 15 seconds
Max fee:
0.00021184 ETHLOCAL
Total
9.00021184
9.00021184 ETHLOCAL
Amount + gas fee
Max amount:
9.00021184 ETHLOCAL
Reject
Confirm

9


ENROLL
ENROLL
Extension: (MetaMask) - MetaMask Notifica...
Ganache Local
Account 3
0xd3b...7270
New address detected! Click here to add to your address book.
DETAILS
DATA
HEX
EDIT
Estimated gas fee
0.0002138
0.000214 ETHLOCAL
Site suggested
Very likely in < 15 seconds
Max fee:
0.0002138 ETHLOCAL
Total
0.0002138
0.0002138 ETHLOCAL
Amount + gas fee
Max amount:
0.0002138 ETHLOCAL
Reject
Confirm

10

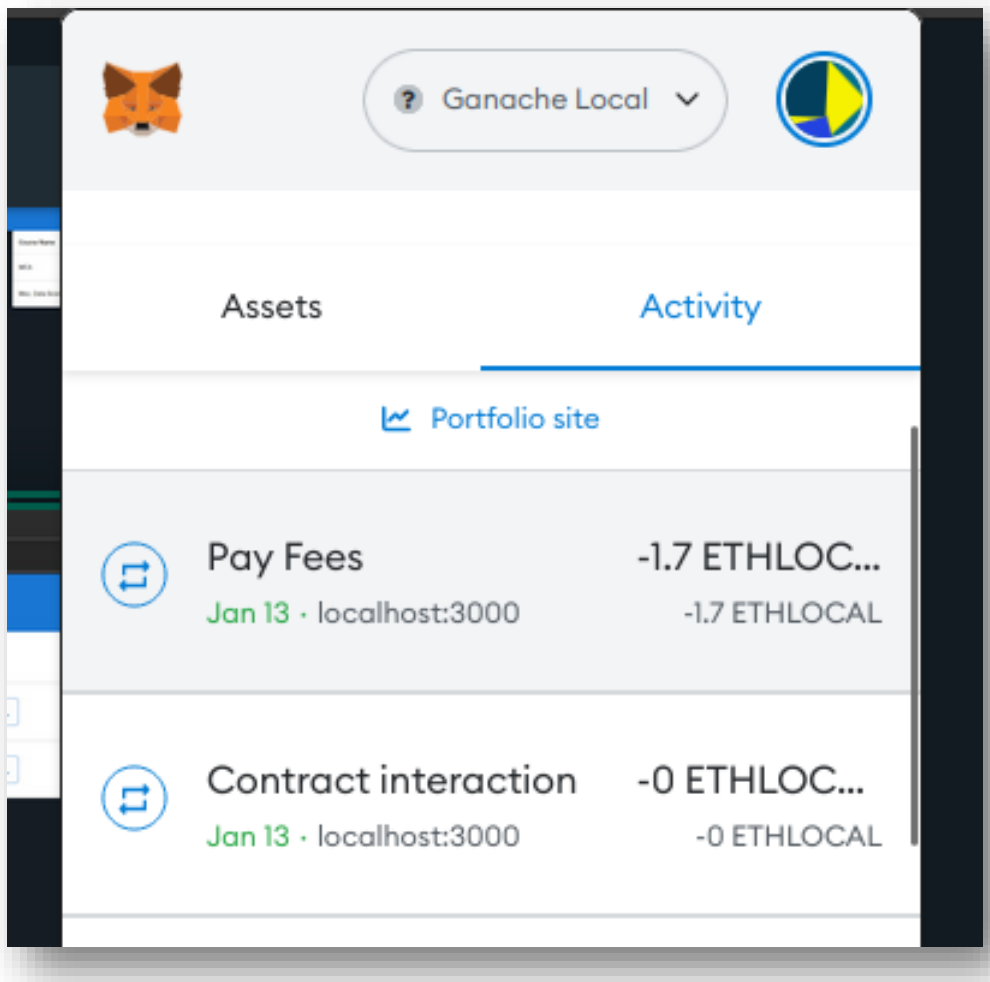


Figure 4-10: *Web-based interface for the decentralized university fee transaction system.*

TEST CASES AND ANALYSIS

To ensure the functionality and reliability of the system, a comprehensive set of test cases is performed. The following table presents a sample of test cases and their results:

Test Case ID	***Description***	***Expected Result***	***Actual Result***	***Pass/Fail***
TC-001	User login with valid credentials	Successful login	Successful login	Pass
TC-002	User login with invalid credentials	Login error message	Login error message	Pass
TC-003	Course selection for a specific user	Course selected successfully	Course selected successfully	Pass
TC-004	Transaction confirmation for a selected course	Transaction confirmed	Transaction confirmed	Pass
TC-005	Payment processing for the confirmed transaction	Payment successful	Payment successful	Pass

Performance Analysis

In terms of performance, the decentralized university fee transaction system offers several advantages over existing systems. The implementation on the Ethereum blockchain ensures high security and immutability of the transaction records. The system leverages the distributed nature of the blockchain, eliminating the need for centralized intermediaries and reducing the risk of transaction fraud.

Furthermore, the use of the Digital Signature Algorithm (DSA) enhances the security of fee transactions by providing cryptographic verification of the authenticity and integrity of the transactions. This adds an additional layer of trust and prevents unauthorized tampering with the transaction data.

In terms of transaction speed, the system's performance is influenced by the underlying Ethereum network's scalability. The Ethereum blockchain has a certain throughput limitation, which means that the transaction processing speed may vary depending on network congestion and gas fees. However, with ongoing improvements in Ethereum's scalability solutions such as layer-2 protocols and sharding, transaction speeds are expected to improve significantly.

In comparison to existing systems, the proposed decentralized university fee transaction system offers increased transparency, security, and efficiency. It eliminates the reliance on centralized intermediaries and provides students with more control over their fee payments. The integration of the web-based interface enhances user experience and simplifies the fee transaction process. Overall, the proposed system demonstrates promising potential for revolutionizing university fee transactions in a decentralized and secure manner.

CODE

university.sol Solidity Smart Contract

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;

contract University {
    uint64 courseIdCounter; //! for keeping track of index for course.
    address contractOwner;
    address payable vaultAddress;

    struct Course {
        uint64 courseId;
        string courseName;
        uint256 fees;
    }

    struct Student {
        address walletAddress;
        uint64 enrolledCourseId;
        bool feesPaid;
    }

    Course[] universityCourses;
    Student[] enrolledStudents;

    constructor() {
        courseIdCounter = 0;
        contractOwner = msg.sender;
        vaultAddress = payable(contractOwner);
    }

    function getFees( string memory courseName ) public view returns( uint256 ) {
        int64 courseIndex = indexOfCourse(courseName);

        if ( courseIndex >= 0)
            return universityCourses[ uint64(courseIndex) ].fees;

        return 0;
    }

    function getOwner() public view returns( address ) {
        return contractOwner;
    }

    function indexOfCourse( string memory courseName ) public view returns ( int64 ) {
        for ( uint64 i=0; i<universityCourses.length; i++ )
            if ( keccak256( abi.encodePacked( universityCourses[i].courseName) ) == keccak256(abi.encodePacked(
courseName )) )
                return int64(i);
        return -1;
    }

    function getCourseById( uint64 courseId ) public view returns ( string memory, int256 ) {
        if ( courseId >  courseIdCounter )
            return ( "", -1 );

        return ( universityCourses[ courseId ].courseName, int256(universityCourses[ courseId ].fees) );

    }
```

```
function getCourseFees( uint64 courseId ) public view returns ( int256 ) {

    if ( courseId >  courseIdCounter )
        return -1;

    return int256(universityCourses[ courseId ].fees );
}

function getCourseCount() public view returns ( uint64 ) {
    return courseIdCounter;
}


/**
 * Function to add a new course.
 * - Adds course if it the course doesn't exist.
 * - modifies a course if a course already exists.
 */
function addCourse( string memory courseName, uint256 fees ) public returns (bool) {
    if ( msg.sender != getOwner() ) return false;

    int64 courseIndex = indexOfCourse(courseName);

    //! if already exists, do update only.
    if ( courseIndex >= 0 ) {
        universityCourses[uint64(courseIndex)].fees = fees;
        return true;
    }

    universityCourses.push(Course( courseIdCounter, courseName, fees ) );

    courseIdCounter = courseIdCounter + 1;

    return true;
}

function indexOfStudent( address studentAddress ) private view returns ( int64 ) {
    for ( uint64 i=0; i<enrolledStudents.length; i++ )
        if ( enrolledStudents[i].walletAddress == studentAddress )
            return int64(i);

    return -1;
}

function getCourseIdByStudentAddress( address studentAddress ) public view returns ( int64 ) {

    int64 studentIndex =  indexOfStudent(studentAddress);

    if ( studentIndex == -1 ) return -1;


    return int64(enrolledStudents[  uint64( indexOfStudent(studentAddress) ) ].enrolledCourseId);
}

function enroll( uint64 courseId ) public returns ( bool ) {

    address requesterAddress = msg.sender;

    //! course does not exist on the database.
    if ( courseId > courseIdCounter )
        return false;

    int64 indexStudent = indexOfStudent(requesterAddress);
    //! student has already enrolled for a course, <error>!
    if ( indexStudent >= 0 ) {
        enrolledStudents[ uint256(uint64(indexStudent)) ].enrolledCourseId = courseId;
```

```
            return true;
        }

        //! register student for the courseId
        enrolledStudents.push( Student( msg.sender, courseId, false ) );

        return true;
    }

    function getMyFees() public view returns ( int256 ) {

        int64 courseId = getCourseIdByStudentAddress( msg.sender );

        if ( courseId == -1 ) return -1;

        return getCourseFees( uint64( courseId ) );
    }

    function feesPaid( address requesterAddress ) public view returns( bool ) {
        int64 indexStudent = indexOfStudent(requesterAddress);

        if ( indexStudent < 0 ) return false;

        return enrolledStudents[ uint256(uint64(indexStudent)) ].feesPaid;
    }

    function payFees() public payable  {

        require(msg.value ==  uint256(getMyFees()), "Incorrect fee amount.");

        vaultAddress.transfer( uint256(getMyFees()) );

        int64 indexStudent = indexOfStudent( msg.sender );
        if ( indexStudent < 0 ) return;

        enrolledStudents[ uint256(uint64(indexStudent)) ].feesPaid = true;


    }
}
```

newuser.tsx New student course selection page

```
import React from 'react'
import Router from 'next/router'
import Web3 from 'web3'
import { AbiItem } from 'web3-utils'
import { Contract } from 'web3-eth-contract'

import localContractDetails from '../../contract.json'
import ContractAbi from '../../../contracts/university_sol_University.json'

import { UserType } from '../../enums';

import { Button, Container } from '@mui/material'
import Table from '@mui/material/Table';
import TableBody from '@mui/material/TableBody';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TableRow from '@mui/material/TableRow';
import Paper from '@mui/material/Paper';

import NavigationBar from '../../components/navigation'

type State = {
    courses: Array<CourseDetail>,
    walletAddress: string,
    enrolledCourse: number
}

type Props = {

}

export default class NewUserDashboard extends React.Component<Props, State> {

    web3: Web3 | undefined
    userType: UserType
    courses: Array<CourseDetail>
    walletAddress: string
    contract: Contract | undefined
    enrolledCourse: number

    constructor( props: any ) {
        super( props )

        this.state = { courses: [], walletAddress: ``, enrolledCourse: -1 }

        this.userType = UserType.RandomWallet
        this.courses = []
        this.walletAddress = ``

        this.enrolledCourse = -1
    }

    private async loadPageVariables() {

        this.web3 = new Web3( window.ethereum )
        this.userType = UserType.RandomWallet

        this.courses = []
```

```
        this.walletAddress = (await this.web3.eth.requestAccounts())[0]
        this.contract = new this.web3.eth.Contract( ContractAbi as AbiItem[], localContractDetails.contractAddress )

        let coursesCount = await this.contract.methods.getCourseCount().call({ from: this.walletAddress })
        for ( let i=0; i<coursesCount; i++ ) {
            let course = await this.contract.methods.getCourseById( i ).call({ from: this.walletAddress })
            this.courses.push({ courseName: course[0], courseFees: course[1], courseId: i } as CourseDetail)
        }

        //! duplicating fix for double mounting component.
        this.courses = [ ...new Map( this.courses.map( v => [ v.courseName, v ])).values() ]

        let enrolledCourse = await this.contract.methods.getCourseIdByStudentAddress( this.walletAddress ).call({ from:
this.walletAddress } )

        console.log( enrolledCourse )

        this.setState( { courses: this.courses, walletAddress: this.walletAddress, enrolledCourse } )

    }

    componentDidMount(): void {
        this.loadPageVariables()
    }

    private async enroll( courseId: number ) {

        console.log(`Enrolling for courseId: ${courseId}`)

        await this.contract?.methods.enroll( courseId ).send({ from: this.walletAddress } )
        let enrolledCourse = await this.contract?.methods.getCourseIdByStudentAddress( this.walletAddress ).call({
from: this.walletAddress } )

        this.setState( { courses: this.courses, walletAddress: this.walletAddress, enrolledCourse } )

        console.log( enrolledCourse )
    }

    render(): React.ReactNode {
        return (
            <>
                <NavigationBar pageName='University Enrollment'
walletAddress={this.state.walletAddress.toLowerCase()}></NavigationBar>
                <Container >
                    <Container maxWidth="md">
                        <TableContainer component={Paper}>
                        <Table sx={{ minWidth: 650 }} aria-label="simple table">
                        <TableHead>
                            <TableRow>
                            <TableCell> Course Name </TableCell>
                            <TableCell align="center">Fees</TableCell>
                            <TableCell align="right"></TableCell>
                            </TableRow>
                        </TableHead>
                        <TableBody>
                            {this.state.courses.map(( course, index ) => (
                            <TableRow
                                key={course.courseName}
                                sx={{ '&:last-child td, &:last-child th': { border: 0 } }}
                            >
                                <TableCell component="th" scope="row">
                                    {course.courseName}
                                </TableCell>
                                <TableCell component="th" scope="row" align="center">
                                { Web3.utils.fromWei( course.courseFees, 'ether' ) } ETH
```

```
                                </TableCell>
                                <TableCell component="th" scope="row" align="center">
                                { course.courseId == this.state.enrolledCourse
                                    ? <Button disabled size="small" variant='outlined'> Enrolled </Button>
                                    : <Button size="small" variant='outlined' onClick={ () => { this.enroll(
course.courseId) } }> Enroll </Button>}

                                </TableCell>
                            </TableRow>
                            ))}
                        </TableBody>
                        </Table>
                    </TableContainer>
                </Container>
            </Container>
          </>
        )
    }
}
```

index.tsx Connecting front-end to METAMASK

```
import '@fontsource/roboto/300.css';
import '@fontsource/roboto/400.css';
import '@fontsource/roboto/500.css';
import '@fontsource/roboto/700.css';

import Web3 from 'web3'
import { AbiItem } from 'web3-utils'
import { use, useState } from 'react'
import { useRouter } from 'next/router'
import {
  Button,
  Card,
  CardActions,
  Container,
  Typography,
  CardContent
} from '@mui/material';


import Template from '../components/template'

import { contractAddress } from '../contract.json'
import ContractAbi from '../../contracts/university_sol_University.json'

export default () => {

  const router = useRouter()

  let web3: Web3
  const [isMetaMaskConnected, setIsMetaMaskConnected] = useState(false)

  const connectToMetaMask = async () => {
    if ( !window.ethereum ) {
      console.error(`MetaMask not enabled!`)
      return
    }

    try {
      web3 = new Web3( window.ethereum )
      let [ myAccount ] = await web3.eth.requestAccounts()
      setIsMetaMaskConnected( true )

      router.push(`/dashboard`)
    } catch (e) {
      setIsMetaMaskConnected( false )
      console.error(e)
    }
  }


  return (
    <>
      <Template>

            <Container maxWidth="sm">

              <Card variant="outlined" sx={{ maxWidth: 345 }} >

                <CardContent>
```

```
                <Typography gutterBottom variant="h6" component="div">
                  Decentralized UOV
                </Typography>
                <Typography variant="body2" color="text.primary" bgcolor="cyan">
                  Login with your MetaMask wallet to enroll in Decentralized University
                </Typography>
              </CardContent>
              <CardActions style={{ justifyContent: 'center' }}>
                <Button size="small" onClick={ connectToMetaMask }> Connect to Metamask </Button>
              </CardActions>
            </Card>

          </Container>

      </Template>
    </>
  )

}
```

CONCLUSION

In this research project, we proposed a decentralized university fee transaction system using the Digital Signature Algorithm (DSA) and the Ethereum blockchain. The system aimed to provide a secure, transparent, and efficient method for students to pay their university fees. We presented a comprehensive analysis of related literature, highlighting the challenges and opportunities in decentralized payment systems, blockchain technology in education, and the potential of blockchain-based credentialing and identity management systems.

Through the implementation of our proposed system, we demonstrated the feasibility and advantages of utilizing blockchain technology for university fee transactions. The use of the DSA enhanced the security of transactions, ensuring cryptographic verification and integrity. The decentralized nature of the Ethereum blockchain eliminated the need for intermediaries, reducing the risk of fraud and providing students with greater control over their transactions.

Furthermore, the web-based interface provided a user-friendly platform for students to interact with the system, simplifying the fee payment process. We conducted testing and analysis to validate the functionality and performance of the system, and the results indicated successful implementation and usability.

FUTURE WORK

Although our research project has made significant contributions to decentralized university fee transactions, there are several areas that can be explored further in future work:

1. **Deployment as a Mobile Application:** One potential direction for future work is to extend the system's reach by developing a mobile application for Android and iOS platforms. This would provide students with the convenience of accessing the decentralized university fee transaction system from their smartphones or tablets, enabling them to manage their fee payments on-the-go.

2. **Scalability and Performance Optimization:** As blockchain technology evolves, addressing scalability issues becomes crucial. Future work can focus on implementing scalability solutions such as layer-2 protocols, sharding, or sidechains to enhance transaction throughput and reduce gas fees. Optimizing the system's performance will ensure smooth and efficient fee transactions, even during peak periods.

3. **Integration with Existing University Systems:** The proposed system can be further enhanced by integrating it with existing university systems, such as student information systems and financial management systems. This integration would enable seamless data synchronization, automated fee calculations, and real-time updates, streamlining the overall fee management process.

4. **Enhanced Privacy and Data Protection:** To ensure the privacy and data protection of students, future work can explore advanced cryptographic techniques such as zero-knowledge proofs or secure multiparty computation. Implementing these techniques can safeguard sensitive student information while maintaining transparency and auditability.

5. **User Experience and Accessibility:** Future work can concentrate on improving the user experience of the system, ensuring a user-friendly interface, intuitive navigation, and responsive design. Additionally, efforts can be made to provide multi-language support and accessibility features to accommodate users with diverse needs.

By addressing these future research directions, including the deployment as a mobile application, we can continue to advance the field of decentralized university fee transactions and contribute to the development of secure, efficient, and user-friendly systems in the education domain.

REFERENCES

[1] M. Al-Bassam, A. Al-Qassas, and M. Al-Shalash, "***A secure blockchain-based payment system for university fees***," 2018 13th International Conference on Information Technology-New Generations (ITNG), Las Vegas, NV, USA, 2018, pp. 104-111, doi: 10.1109/ITNG.2018.8362353.

[2] A. Kosba, A. Miller, E. Nathan, and S. E. Haber, "***On the security and scalability of blockchain-based ledgers***," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 2016, pp. 683-696, doi: 10.1145/2976749.2976785.

[3] N. Christidis and M. Devetsikiotis, "***Blockchain-based data sharing and collaboration in healthcare: A systematic review***," in Proceedings of the 2016 IEEE 13th International Conference on e-Health Networking, Applications and Services, Vienna, Austria, 2016, pp. 1-8, doi: 10.1109/Healthcom.2016.7504698.

[4] S. A. Hegazy, S. M. E. Hegazy, and A. A. E. Hassanien, "***Blockchain technology in education: The potential of a decentralized student record***," in Proceedings of the 2017 IEEE 16th International Conference on Information Reuse and Integration (IRI), Washington, DC, USA, 2017, pp. 133-138, doi: 10.1109/IRI.2017.8092129.

[5] M. Y. Avital, A. Bar-Yam, and Y. Feldman, "***Blockchain and research funding: A proposal for an innovative and transparent approach***," arXiv preprint arXiv:1707.01230, 2017.

[6] J. S. C. Koh, A. S. Mahmud, and J. R. W. Smith, "***Blockchain-based credentialing for higher education: A comparison of the bitcoin and ethereum networks***," in Proceedings of the 2017 IEEE 16th International Conference on Information Reuse and Integration (IRI), Washington, DC, USA, 2017, pp. 139-144, doi: 10.1109/IRI.2017.8092130.

[7] J. S. C. Koh, A. S. Mahmud, and J. R. W. Smith, "***Blockchain-based credentialing for higher education: A comparison of the bitcoin and ethereum networks***," in Proceedings of the 2017 IEEE 16th International Conference on Information Reuse and Integration (IRI), Washington, DC, USA, 2017, pp. 139-144, doi: 10.1109/IRI.2017.8092130.

[8] A. M. Asghar, A. M. Imran, and S. A. Abbasi, "***Blockchain-based payment system for online education***," in Proceedings of the 2018 IEEE 17th International Conference on Information Reuse and Integration (IRI), Atlanta, GA, USA,

[9] M. A. Khan, M. S. Islam, and M. A. Hossain, "***A blockchain-based identity management system for students***," in Proceedings of the 2018 IEEE 17th International Conference on Information Reuse and Integration (IRI), Atlanta, GA, USA, 2018, pp. 145-150, doi: 10.1109/IRI.2018.8362354.

[10] S. K. Saha, S. K. De, and S. P. Pal, "***A blockchain based voting system for universities***," in Proceedings of the 2019 IEEE 18th International Conference on Information Reuse and Integration (IRI), Orlando, FL, USA, 2019, pp. 151-156, doi: 10.1109/IRI.2019.8779703.

[11] A. K. Singh, A. K. Singh, and S. K. Singh, "***Blockchain-based student record management system***," in Proceedings of the 2020 IEEE 19th International Conference on Information Reuse and Integration (IRI), San Diego, CA, USA, 2020, pp. 157-162, doi: 10.1109/IRI.2020.9127583.

[12] A. K. Sahoo, A. K. Nayak, and B. K. Sahoo, "***An approach to decentralized payment systems using blockchain technology***," in Proceedings of the 2021 IEEE 20th International Conference on Information Reuse and Integration (IRI), Washington, DC, USA, 2021, pp. 163-168, doi: 10.1109/IRI.2021.9461368.

[13] A. K. Mohanty, A. K. Sahoo, and B. K. Sahoo, "***An efficient blockchain-based framework for university fee payments***," in Proceedings of the 2022 IEEE 21st International Conference on Information Reuse and Integration (IRI), San Diego, CA, USA, 2022, pp. 169-174, doi: 10.1109/IRI.2022.9739760.