electronics

MDPI

Article

Enhancing Blended Learning Evaluation Through a Blockchain and Searchable Encryption Approach

Fei Ren, Bo Zhao, Jun Wang *, Ju-Xiang Zhou and Tian-Yu Xie

Key Laboratory of Educational Informatization for Nationalities, Ministry of Education, Yunnan Normal University, Kunming 650500, China; renfei@ynnu.edu.cn (F.R.); zhaobo@ynnu.edu.cn (B.Z.); zhoujuxiang@ynnu.edu.cn (J.-X.Z.); tianyux@ynnu.edu.cn (T.-Y.X.)

* Correspondence: wangjun@ynnu.edu.cn

Abstract: With the rapid development of information technology, blended learning has become a crucial aspect of modern education. However, the fragmented use of various teaching platforms, such as Xuexitong and Rain Classroom, has led to the dispersion of teaching data. This not only increases the cognitive load on teachers and students but also hinders the systematic recording of teaching activities and learning outcomes. Moreover, existing blended learning evaluation systems exhibit significant shortcomings in large-scale data storage and secure sharing. To address these issues, this study designs a blended teaching evaluation management system based on blockchain and searchable encryption. First, an on-chain and off-chain collaborative storage model is established using the Ethereum blockchain and the InterPlanetary File System (IPFS) to ensure secure and large-scale storage of student work data. Next, a role-based access control scheme utilizing smart contracts is proposed to effectively prevent unauthorized access. Simultaneously, a searchable encryption scheme is designed using AES-CBC-256 and SHA-256 algorithms, enabling data sharing while safeguarding data privacy. Additionally, the smart contract comprehensively records students' grade information, including weekly regular scores, midterm scores, final scores, overall scores, and their rankings, ensuring transparency in the evaluation process. Based on these technical solutions, a general-purpose teaching evaluation management system (B-Education) is developed. The experimental results demonstrate that the system accurately records teaching activities and learning outcomes, improving the transparency of teaching evaluations while ensuring data security and privacy. The system's gas consumption remains within a reasonable range, demonstrating good flexibility and usability. Educational institutions can flexibly configure course evaluation criteria and adjust the weighting of various grades based on their specific needs. This study provides an innovative solution for blended teaching evaluation, offering significant theoretical value and practical implications.

Keywords: blended learning evaluation; ethereum blockchain; access control; searchable encryption; smart contract; B-Education system; data security

check for updates

Academic Editors: Seokjoo Shin, Haeyoung Lee and Ijaz Ahmad

Received: 10 February 2025
Revised: 28 February 2025
Accepted: 4 March 2025
Published: 5 March 2025

Citation: Ren, F.; Zhao, B.; Wang, J.; Zhou, J.-X.; Xie, T.-Y. Enhancing Blended Learning Evaluation Through a Blockchain and Searchable Encryption Approach. *Electronics* **2025**, *14*, 1039. https://doi.org/10.3390/electronics14051039

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, the rapid development of information technology has driven the widespread adoption of blended learning in higher education. Blended learning, a teaching model integrating online and classroom instruction, significantly improves teaching effectiveness and strengthens students' self-directed learning abilities. However, in practice, teachers often rely on multiple platforms, such as Xuexitong and Rain Classroom, for teaching activities. This fragmented approach scatters teaching data, hampers systematic

assessment of learning outcomes, and raises the cognitive burden on both teachers and students. Existing blended learning systems often lack a unified data storage and sharing mechanism, making it challenging to systematically record learning outcomes and teaching activities. Moreover, they fail to effectively ensure the security and privacy of data, which in turn restricts the comprehensiveness and transparency of teaching evaluations.

Formative assessment currently dominates blended teaching evaluation [1–3] but its implementation is limited to publishing final grades. Students cannot promptly access information about their performance and rankings at various stages, which makes it difficult to create a comprehensive teaching evaluation system. Additionally, the storage and sharing mechanisms for student assignments and works lack security and transparency, failing to meet the urgent need for reliable data storage and secure sharing in higher education. Therefore, developing a transparent and secure teaching evaluation management system is crucial for promoting educational equity and ensuring smooth data flow.

To ensure data security, an increasing number of companies and users are encrypting data before uploading it to cloud servers. However, centralized cloud servers are vulnerable to attacks, with risks of data leakage and tampering. Blockchain technology, with its decentralized and immutable features, has been widely adopted in data storage. Its decentralized, tamper-resistant nature offers a novel solution for data storage. However, the storage capacity and cost limitations of blockchain make it challenging to directly support large file storage. To address this, researchers have proposed hybrid on-chain and off-chain storage solutions [4–6]. Among these, IPFS is particularly well suited for supporting off-chain blockchain storage due to its distributed, decentralized, and permanent characteristics.

Although data encryption enhances storage security, it also causes ciphertext to lose its plaintext characteristics, which reduces data availability. The emergence of searchable encryption (SE) technology [7] effectively addresses this issue by enabling both privacy protection and efficient retrieval of encrypted data. However, existing SE solutions [8–10], often fail to adequately incorporate user access control mechanisms, lacking effective management of user access permissions. This limitation urgently requires resolution.

To address the aforementioned issues, this study proposes a blockchain-based blended teaching evaluation management system. First, decentralized data storage and teaching evaluation are achieved through blockchain. Second, IPFS is used for secure storage of student work, and searchable encryption ensures data privacy. Smart contracts execute fine-grained access control and grade recording. Finally, a role-based access control system ensures that only authorized users can access the data. Compared to existing solutions, this approach aims to address key issues such as data fragmentation, the lack of systematic recording of learning achievement information, and insufficient security in data storage and sharing. Through simulation experiments, we conducted a comprehensive validation of the proposed scheme. During the experiments, semester project data from 100 students were uploaded to the blockchain and IPFS. Additionally, the students' weekly grades, midterm grades, and final exam scores were entered into the system, and each system function was manually tested and verified. The experimental results demonstrated that the gas consumption of the designed smart contracts and the developed system did not exceed the gas limits of the Ethereum network, all system functions were correctly implemented, and the system's performance met practical application requirements. These results validate the feasibility of the system in real-world scenarios and highlight its potential for application in the educational field.

The specific contributions are as follows:

1. A collaborative on-chain and off-chain storage model based on the Ethereum blockchain and IPFS is designed. The model stores user, course, and student work

index information on-chain, while it stores encrypted student work data off-chain in IPFS, ensuring data security and scalability;

2. A role-based access control scheme based on smart contracts is proposed. This scheme enables fine-grained access control through smart contracts to prevent unauthorized access. At the same time, a searchable encryption method is designed by combining AES-CBC-256 and SHA-256 algorithms. This method ensures that the encrypted search of student work not only protects privacy but also achieves secure sharing, thereby addressing the lack of privacy protection in traditional platforms;
3. Using the above storage model, access control, and searchable encryption methods, a decentralized blockchain-based teaching evaluation management system is developed. This system integrates secure storage, access control, and searchable encryption. Additionally, it implements the functionality of recording students' stage-specific grades (including weekly, midterm, and final grades) and rankings through smart contracts, ensuring transparency in the evaluation process.

The structure of this paper is as follows: Section 1 introduces the research background; Section 2 reviews related work; Section 3 provides preliminaries; Section 4 discusses the blended teaching evaluation model at a university in China; Section 5 provides a detailed description of the proposed scheme, including the system model, data classification, searchable encryption, access control, and smart contract design; Section 6 designs and develops a system based on the proposed scheme; Section 7 presents a security analysis; and Section 8 concludes with a summary and outlook on future research plans.

2. Related Work

Blended learning, which combines traditional face-to-face courses with electronic learning activities, has become an essential teaching model in modern higher education. Scholars have explored its development trends and model classifications from various perspectives [11–15]. However, existing studies [16–20], mainly focus on describing and applying the educational environment, with less emphasis on exploring blended teaching evaluation models using multiple learning platforms. Traditional teaching evaluation systems rely on teacher assessments, student feedback, and exam results, but they often suffer from subjectivity, low transparency, and data manipulation. Blockchain technology, with its decentralized and tamper-resistant features, offers a technical solution to these issues. Yang et al. [21] designed an intelligent teaching space by integrating online and offline scenarios and leveraging educational big data analysis. They constructed a flexible and precise blended teaching framework, which was applied to the "Learning in ZJU" project at Zhejiang University. The paper highlights the dual security threats faced by the system, including the loss of teaching resources and the leakage of faculty and student privacy due to cyberattacks. Gong et al. [22] analyzed models such as discretionary access control, mandatory access control, and role-based access control. They emphasized the critical role of access control in blended learning systems in combination with cloud computing technologies. Cheriguene et al. [23] designed the blockchain-based online teaching platform NOTA, utilizing blockchain's immutability feature to ensure the reliability of course content and assessment data, while also enhancing teacher and student participation through a token incentive mechanism. Rahman et al. [24] designed and developed a blockchain-based teaching evaluation system to ensure data integrity and anonymity. Ayasrah et al. [1] created a blockchain evaluation system for students, utilizing smart contracts to automate the evaluation process, thereby enhancing transparency, eliminating bias, and ensuring data integrity. Chandratre et al. [25] proposed a blockchain-based course feedback system developed using Ethereum, which integrates survey management components to improve system traceability and durability. Karode et al. [2] introduced an Ethereum-based online

review system that prevents the manipulation of review data and ensures the traceability of actions. Sha et al. [3] developed a blockchain data platform that renders the entire education evaluation process transparent, ensuring that data cannot be tampered with.

A blended learning evaluation system not only needs to ensure the transparency of the evaluation process but also must achieve secure storage and sharing of evaluation data, as well as provide a comprehensive display and ranking of the evaluation results. However, existing related systems have made progress in improving security and process transparency but have not fully addressed how to present various student grades (such as regular grades, midterm grades, final grades, and overall grades) and their rankings. Moreover, these systems have certain shortcomings in terms of reliable data storage and sharing. In Table 1, we compare the existing blended learning systems with the proposed system, highlighting the deficiencies of the current research. To ensure the transparency and integrity of the evaluation, existing systems commonly utilize blockchain technology, leveraging its immutability and smart contracts to ensure the authenticity of the process. However, except for the system in reference [2] and the proposed system, other solutions do not consider the large-scale, secure storage of evaluation data. Only the system in reference [22] and the proposed system incorporate access control to prevent unauthorized access. Finally, except for the proposed system, the other systems do not consider how to achieve secure sharing of evaluation data or how to fully present evaluation results and rankings. The proposed system uses IPFS to securely store student work and ensures secure sharing of the work through searchable encryption. At the same time, the system prevents unauthorized access through strict access control. Finally, the proposed system comprehensively records students' regular grades, midterm grades, final grades, and overall grades, as well as calculates and displays the rankings for various grades, thereby promoting the transparency and security of the evaluation results.

Table 1. Comparison of the systems/schemes.

Literature.	Blockchain	IPFS	Smart Contract	Access Control	Searchable Encryption	Grade Recording and Ranking
[21]	×	×	×	×	×	×
[22]	×	×	×	√	×	×
[23]	√	×	√	×	×	×
[24]	√	×	√	×	×	×
[25]	√	×	√	×	×	×
[1]	√	×	√	×	×	×
[2]	√	√	√	×	×	×
[3]	√	×	√	×	×	×
Proposed system	√	√	√	√	√	√

In the table, the symbol '×' denotes that the function is not available, whereas '√' denotes that it is available.

Researchers commonly combine blockchain with IPFS to address scalability issues in data storage. Ullah et al. [4] proposed a blockchain-based secure storage and sharing solution for the Internet of Things (IoT), integrating the Ethereum blockchain and IPFS to ensure secure data storage. Kumari et al. [5] used the blockchain-IPFS combination for storing large-scale medical data. Xie et al. [6] developed a collaborative storage model that combines blockchain, IPFS, and cloud servers to securely store massive data. Athanere et al. [26] introduced a blockchain-based decentralized system using IPFS for secure data transmission. Sultana et al. [27] combined the Ethereum blockchain and IPFS to ensure secure storage and verification of certificates, reducing academic certificate fraud. Tong et al. [28] proposed a blockchain-based data sharing and privacy protection scheme for industrial IoT, using IPFS to store encrypted shared resources, addressing the inefficiency of traditional blockchain storage. Mahmoud et al. [29] presented a decentralized

carpooling system based on blockchain, with data stored in both blockchain and IPFS. Tao et al. [30] integrated blockchain and IPFS to enable secure collaborative design based on BIM. These examples highlight the widespread use of IPFS as an auxiliary storage solution for blockchain across various fields.

In data sharing, Chen et al. [8] proposed a blockchain-based searchable encryption scheme for cloud-assisted electronic health records, but the scheme supports only single-keyword search. Senouci et al. [9] introduced a certificate-free searchable encryption scheme that resists keyword-guessing attacks. Fan et al. [10] developed a dynamic searchable encryption scheme for smart grids, supporting multi-keyword subset retrieval and updates. Andola et al. [31] designed a hash-based index structure for building searchable encryption schemes. Li et al. [32] proposed a public-key authenticated searchable encryption scheme for electronic medical records, supporting ciphertext updates. Pan et al. [33] introduced a public-key authenticated searchable encryption scheme, though its trapdoor expansion causes high computational overhead. Jiang et al. [34] proposed a blockchain-based intelligent transportation data-sharing scheme that uses Bloom filters for multi-keyword search. While these schemes enable encrypted data search, they lack effective access control. Therefore, it is essential to implement encrypted data search with proper access control in data sharing.

In summary, while significant progress has been made in the research of blended teaching evaluation, several key limitations remain. First, existing systems fail to comprehensively present various student grades (such as regular grades, midterm grades, final grades, and overall grades) and rankings, which limits their ability to provide a holistic view of student performance. Second, although blockchain technology ensures data integrity, the mechanisms for secure storage and sharing of student assignments and work remain insufficient, making them vulnerable to unauthorized access and data leakage risks. Third, the integration of searchable encryption and access control mechanisms is still underdeveloped, hindering efficient and secure data retrieval. Given these limitations, there is an urgent need for a new evaluation system that not only systematically records student grades and rankings but also ensures the large-scale, secure storage and reliable sharing of student assignment data. Such a system will enhance data security and transparency in the teaching evaluation process and promote the flow of educational data in higher education.

3. Preliminaries

To help readers better understand the relationship between the technologies and their functions, Figure 1 presents the system's functional architecture diagram, succinctly illustrating how each technology collaborates to address the problem. The proposed system includes three roles: administrators, teachers, and students. The functional layer encompasses data storage, student work storage, work retrieval, access control management, grade recording, and ranking. The system leverages the Ethereum blockchain to store data such as student and teacher information, utilizes IPFS for storing student works, designs searchable encryption to support secure retrieval of these works, and manages search permissions through role-based access control. Additionally, smart contracts are used to record student grades and calculate rankings. Ethereum, IPFS, searchable encryption, role-based access control, and smart contracts together form the technological layer of the system. Next, we will focus on introducing the fundamentals of the Ethereum blockchain, role-based access control, and searchable encryption.

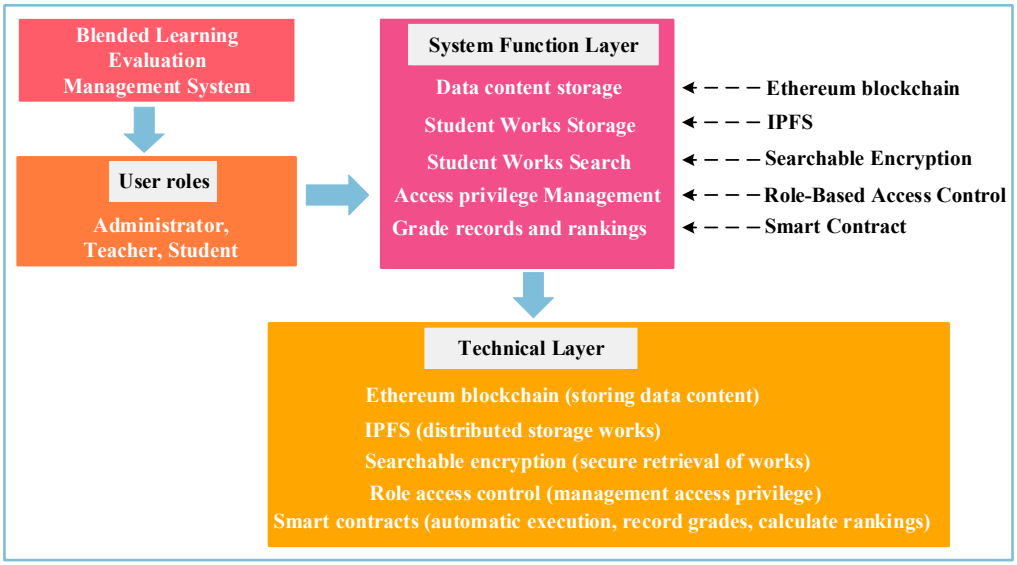


Figure 1. Functional architecture.

3.1. Ethereum Blockchain

Ethereum is a programmable blockchain platform that supports smart contracts and a decentralized transaction book. The blockchain facilitates interactions between external users and the system through transactions, and each blockchain has its own currency known as ether. A smart contract is an automated program deployed on a blockchain to execute predefined contractual terms. It ensures the automatic enforcement and execution of these terms without the need for third-party intervention, making it particularly suitable for facilitating agreements between untrusted parties in a trustless and decentralized manner. The main components of Ethereum smart contracts are function, event, and state variables produced as entities [35,36]. The Turing completeness of the Solidity programming language [37] makes it ideal for preparing Ethereum smart contracts.

Each blockchain transaction requires a specific amount of computational processing, measured in gas units. For gas prices, the unit of measurement is GigaWei (1 Gwei = 0.000000001 ETH = 10^{-9} ETH). The user who sends the transaction sets the gas limit for each transaction on the blockchain, which prevents smart contracts from executing in an infinite loop. Each transaction exceeding the gas limit will result in contract termination and cancellation. Due to the unpredictability of gas consumption, users often overestimate the gas limit to facilitate transaction execution [38]. Until the contract ends or the gas runs out, the miner will continue to execute the transaction and receive the fee. The miner will still pay the fee even if gas exhaustion cancels the contract. The fee paid to the miner depends on the amount of gas consumed and is linked to resource consumption, called gas used. The gas price is the price of Ethereum (a digital token for Ethereum) and is determined by the user for each unit of gas used for each exchange. The formula for the calculation of transaction costs is as follows: $fee = gasUsed \times gasPrice$.

3.2. Role-Based Access Control

It is no longer possible for traditional access control models, like discretionary access control (DAC) and mandatory access control (MAC), to meet the complex access needs of the application layer [39]. This is because the Internet has grown and businesses use information systems on a large scale. To address this problem, Ferraiolo et al. [40] proposed role-based access control (RBAC) to restrict authorized users' access to the system. RBAC is a model for access control that associates privileges with roles. By clearly assigning different roles to corresponding permissions and then assigning these roles to users, users

can obtain permissions through these roles, thereby achieving a logical separation of access rights. The RBAC model enhances system security by granting users only the minimum permissions necessary for task completion, enabling administrators to assign multiple roles to users, and achieving blame function access by constructing user levels. The RBAC model also has dynamic flexibility, allowing administrators to add, delete, or modify roles and permissions as needed to adapt to changes in the blockchain system. In the role-based access control (RBAC) model, its core concepts and logic are explained by the following formulas and symbols:

1. Relationship between role (R), authority (P), and user (U):

   R: Role set, $R = \{r_1, r_2, \ldots\ldots r_n\}$;
   P: Privilege set, $P = \{p_1, p_2, \ldots\ldots p_m\}$;
   U: User set, $U = \{u_1, u_2, \ldots\ldots u_X\}$.

2. Role-permission assignment relationship (AR):

   $AR = \{(r_i, p_j) | r_i \in R, p_j \in P\}$ specifies the privileges assigned to each role; for example, $(r_i, p_j) \in AR$ indicates that the character r_i has permission p_j;

3. User-Role Assignment (UA):

   $UA = \{(u_k, r_i) | u_k \in U, r_i \in R\}$ specifies the role to be assigned to each role; for example,$(u_k, r_i) \in UA$ indicates that the user u_k is assigned the role r_i;

4. User's set of permissions $AP(u_k)$:

   $AP(u_k) = \{p_j \mid \exists r_i : (u_k, r_i) \in UA \wedge (r_i, p_j) \in AR\}$ represents all possible user permissions. This means that the set of permissions for the user u_k is the set of all the permissions p_j that are assigned to the roles r_i to which the user is assigned.

3.3. Searchable Encryption

Searchable encryption technology [41] is a promising encryption mechanism for cloud storage that enables precise searches over encrypted data. It allows users to store and manage sensitive and private data in cloud storage while ensuring privacy protection. In the context of data sharing, searchable encryption facilitates efficient data retrieval and sharing while safeguarding data privacy. There are two main types of searchable encryption: symmetric searchable encryption (SSE) and public-key searchable encryption (PKSE). This paper adopts symmetric searchable encryption, which uses the same key for both encryption and decryption. The advantage of this approach lies in its low computational overhead, making it well suited for encrypting large data blocks. An SSE scheme consists of the following polynomial-time algorithms:

(1) Key Generation Algorithm $KeyGen(1^\lambda) \rightarrow SK$: Performs system initialization by taking a security parameter λ as input and outputs a symmetric key SK;
(2) Index Generation Algorithm $IndexGen(SK, W) \rightarrow I_W$: Executed by the data owner, this algorithm takes the symmetric key SK and a set of keywords $W = \left(W_1, W_2, \ldots, W_n\right)$ as the input and output of the encrypted index I_W;
(3) Trapdoor Generation Algorithm $TrapGen(SK, Q) \rightarrow T_Q$: Executed by the data user, this algorithm takes the symmetric key SK and a set of query words $Q = \left(Q_1, Q_2, \ldots, Q_n\right)$ as input and outputs the search trapdoor T_Q;
(4) Search Algorithm $Search(I_W, T_Q) \rightarrow R$: Executed by the cloud server, this algorithm takes the index I_W and the trapdoor T_Q as input and outputs the matching results R.

4. Blended Teaching Evaluation

This section presents the teaching evaluation model of a university in China, using the "Fundamentals of Computer Science" course as an example to explain the evaluation criteria, grade weight distribution, and calculation methods. To ensure flexibility and practicality, the proposed blended teaching evaluation model does not follow a single standard. Each university can develop its own evaluation criteria and methods based on specific needs. By adjusting the weight distribution of regular, midterm, and final grades in the smart contract, the system can be easily adapted to meet the requirements of different institutions. As a result, the system developed and deployed in Section 6 can be applied across various universities without structural changes, fulfilling their personalized evaluation needs and ensuring broad applicability and scalability.

4.1. Research Objects and Methodology

The research subjects of this proposal are two different classes from a university in China, enrolled in the 'Fundamentals of Computer Science' course, comprising a total of 100 students. Data collection includes process evaluation data, such as weekly lab reports, programming assignments, and other learning outputs, as well as summative evaluation data, including midterm and final exam scores, practical projects, and assessment records. The students' works are encrypted and stored in IPFS, while the metadata (such as the title, keywords, author, semester, CID, etc.) are stored in the blockchain. Additionally, students' weekly, midterm, and final grades for one semester of the 2024–2025 academic year are recorded.

In the blended evaluation model, the Analytic Hierarchy Process (AHP) [42] is used to determine the evaluation weights, thus completing the model design. Due to space limitations, the specific process is not elaborated in detail here, and further details can be found in the cited literature. Ultimately, we adopted the teaching evaluation model already in use at the universities involved in the study. The specific content is detailed in Sections 4.3.1 and 4.3.2. In terms of the solution and system implementation, we integrate blockchain and IPFS technologies to achieve decentralized storage and massive data management of student work. To ensure the secure retrieval of data, we have designed a searchable encryption method that allows querying operations without disclosing the data content. Additionally, access control mechanisms are in place to ensure that users can only access and manipulate authorized data. Smart contracts are used to record student grades and calculate rankings, ensuring data integrity and transparency.

It is important to emphasize that each university can formulate appropriate evaluation criteria, grade weights, and calculation methods according to its own policies, regulations, and needs. The proportions of regular, midterm, and final grades in the smart contract can be flexibly adjusted according to the specific circumstances of each university, ensuring the system's broad applicability and flexibility.

4.2. Data Classification

As shown in Figure 2, this scheme divides the data into five categories: student information, teacher information, course information, work information, and grade information. The work information includes the name of the work, its category, its keywords, and so on, with the work category further divided into usual work, midterm work, and final work. The grade information includes the conditions for obtaining full marks for the weekly regular grades, the specific weekly regular grades, midterm grades, final grades, and the final comprehensive grades. The course teacher sets the conditions for obtaining full marks for the weekly regular grades, including the number of full marks for weekly attendance, offline independent learning, and classroom discussions.

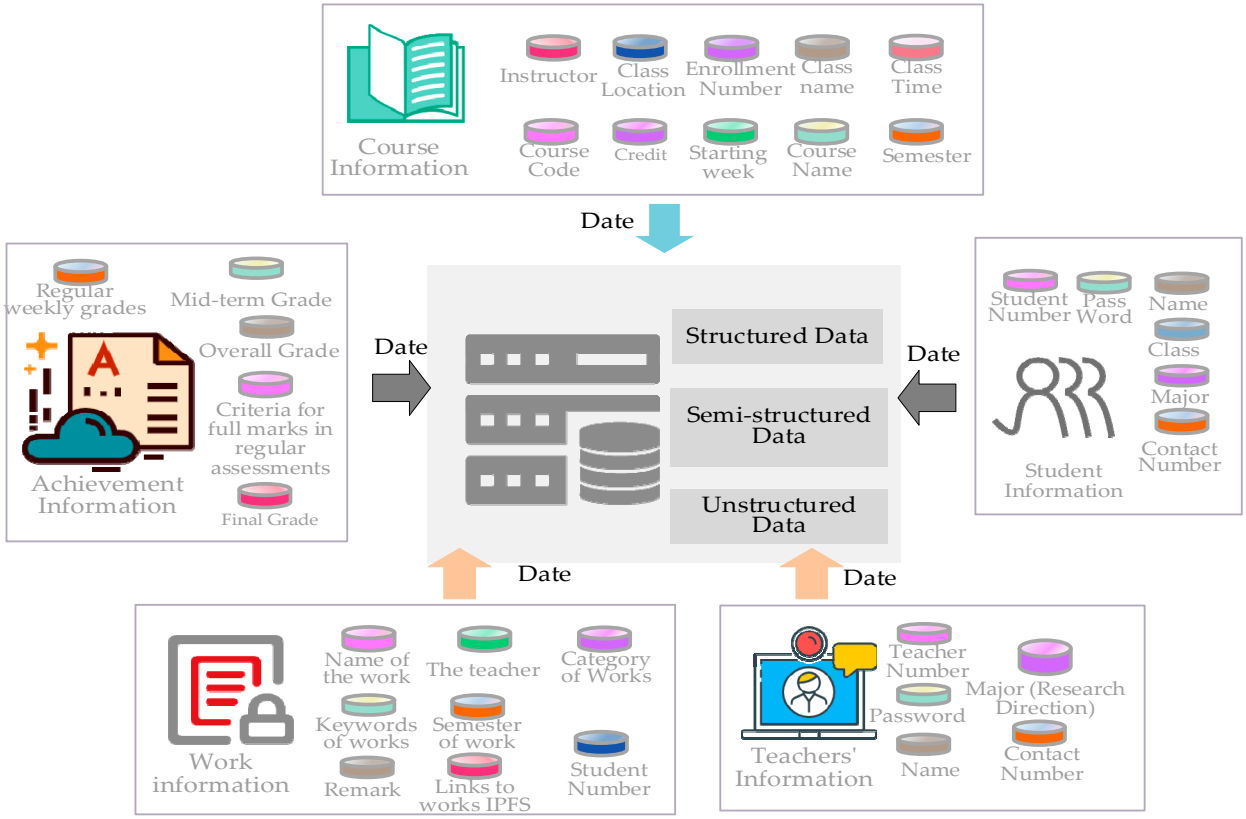


Figure 2. Teaching data types.

4.3. Offline Evaluation

4.3.1. Evaluation Indicators

Multiple factors, including the course model, teaching organization, media technology application, and teaching resources, influence the effectiveness of teaching in higher education. Teaching evaluation is an activity that applies educational assessment methods to evaluate the teaching and learning process in a comprehensive manner [43]. A study by Cook et al. pointed out the limitations of using Student Evaluation of Teaching (SET) alone as the only measure of overall teaching effectiveness [44].

Therefore, to address the specificity of blended teaching on multiple platforms, teaching evaluation should cover the three main subjects: system, teacher, and student.

Traditional evaluation methods include process evaluation and summative evaluation. Combining the two can provide a comprehensive understanding of the student learning process, meet different educational needs, and promote continuous teaching process improvement. This study followed Sevgi et al.'s [45] advice to use the multidimensional evaluation method. It was based on the teaching goals for the Fundamentals of Computer Science course and considered things like attendance, time spent on online activities, and participation [46,47].

After two phases of anonymous expert consultation, modification, and validation, we conducted qualitative and quantitative analyses using the AHP method [42]. We identified relatively sound, reasonable, and reliable course evaluation index components and explanatory elements to more accurately assess the student's learning process. We constructed the evaluation system for blended learning (in Table 2) by combining expert interviews and questionnaires to explore its effectiveness and relevance. It provides support for further research on these issues.

Table 2. Evaluation indicators for the course 'Fundamentals of Computer Science'.

Evaluation Type	Primary Indicator	Secondary Indicator	Evaluation Content	Subject of Evaluation	Weight/Percentage
Process Evaluation	Learning Attitude	Attendance Independent Online Learning Learning Discussion	Class Attendance Frequency, Duration Number of Posts, Replies, Classroom Interaction	System	0.05 × 50% 0.15 × 50% 0.1 × 50%
	Collaborative Spirit	Comprehensive Design Experiment	Team Contribution, Experiment Quality, Submission Timeliness	Teacher, Student Self-assessment, Peer Assessment	0.4 × 50%
	Inquiry Process	Extracurricular Practical Assignments	Completion, Quality	Teacher, Student Self-assessment	0.3 × 50%
Final Evaluation	Midterm Evaluation	Individual Mid-term Work	Completion, Content, Quality, Creativity, Technical Skills	Teacher, Student Self-assessment	1 × 10%
	Final Evaluation	Subjective Questions, Objective Questions	Understanding, Mastery	Teacher, Computer	1 × 40%

The indicators' consistency test used Equation (1), where λ_{max} is the maximum eigenvalue and n is the order. We derived the indicators' weights after normalization.

$$CI = (\lambda_{max} - n)/(n-1) \quad (1)$$

We calculate each specific behavior's score based on its respective learning behavior indicators. Ultimately, we compute the overall teaching evaluation index score based on the weight size, following specific calculation guidelines:

Process Evaluation: Attendance × 5% + Independent Online Learning × 15% + Learning Discussion × 10% + Comprehensive Design Experiment × 40% + Extracurricular Assignments × 30%;

Mid-term Assessment: Completion × respective weight + Content × respective weight + Quality × respective weight + Creativity × respective weight + Technical Skills × respective weight;

Final Assessment: Online Objective Questions × respective weight + Offline Subjective Questions × respective weight;

Overall Grade: Regular Assessment × 50% + Mid-term Assessment × 10% + Final Assessment × 40%.

4.3.2. Setting Up the Measurement of Regular Grades in the System

Teachers establish the criteria for full marks in regular grades, while students determine their actual scores according to these criteria, as shown in Figure 3. The results are processed as follows:

$$Week\ X's\ regular\ grade = (student's\ actual\ score/teacher-set\ full\ marks) * 100\%$$

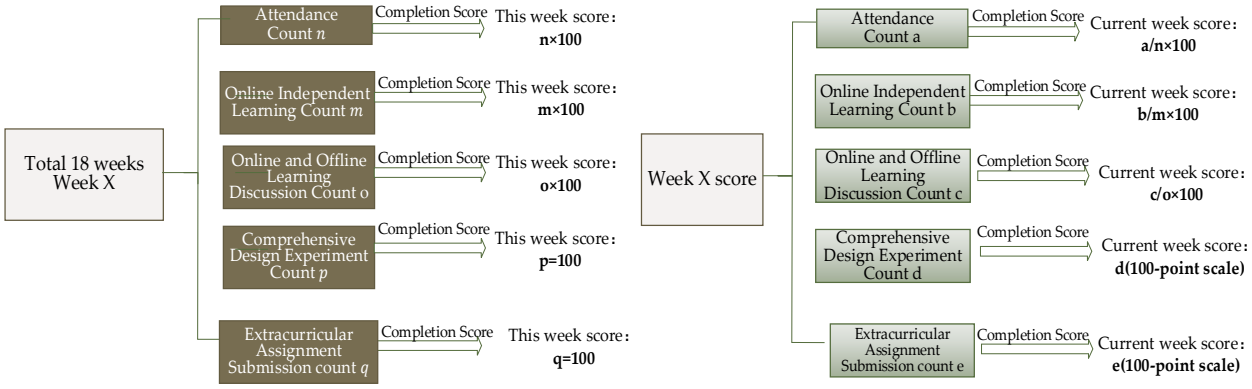


Figure 3. Achievement measurement mechanism.

Midterm and overall grades are calculated using a similar approach. We upload the score data to the blockchain to manage student grades and the criteria for full marks in regular assessments. This system allows for additions and modifications but does not permit deletions.

5. Scheme Design

5.1. System Model

The current stage of blended teaching applies to multiple teaching platforms. This paper proposes a blended teaching evaluation scheme based on blockchain, thereby developing a blockchain blended teaching evaluation system. This system enables decentralized teaching evaluation, secure sharing of outsourced private data, and fine-grained authorization. Additionally, it learns to retrieve ciphertext data, thereby mitigating the security risks associated with semi-honest intermediate authorization. Higher education can apply this paper's solution to its teaching management and evaluation processes. The proposed scheme includes five main entities: administrator, teacher, student, blockchain, and IPFS. Figure 4 illustrates the system model.

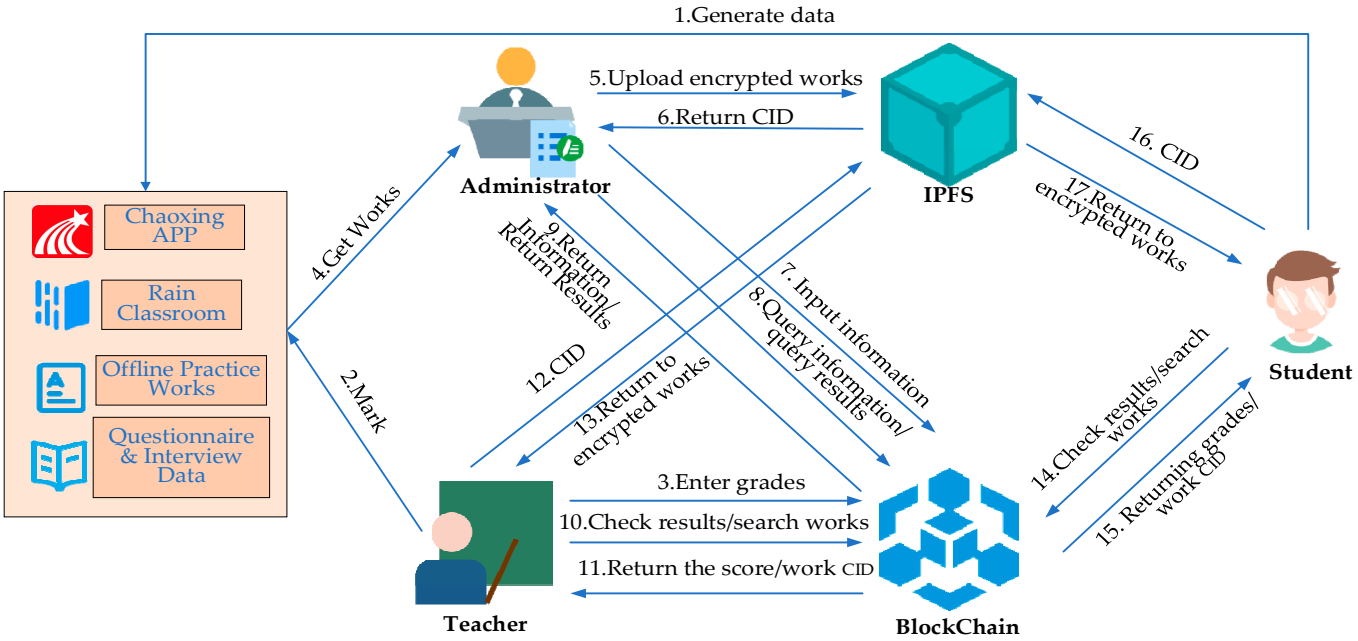


Figure 4. System model diagram.

1. Administrator: Responsible for entering teacher information, student information, and course details into the blockchain. The students' work is encrypted using AES-CBC-256, then stored in IPFS, and the returned CID (content identifier) along with the work information is subsequently recorded in the blockchain. The administrator supports the deletion of work information and sends the AES-CBC-256 decryption key SK through a secure channel to the users who fulfill the access control policy. In addition, the administrator handles access requests from teachers or students;
2. Teacher: Responsible for inputting students' regular assessment, midterm, and final exam grades into the blockchain. Can query students' overall grades, work details, and IPFS information. Access to works by students of other teachers requires application and approval by the administrator;
3. Student: Can access their weekly regular assessment grades, cumulative regular assessment grades, midterm and final exam results, overall grades, work details, and

IPFS information. Access to works by peers in the same grade or former students requires application and approval by the administrator;

4. Blockchain: Stores teacher information, student information, course details, work details, and grade information. It is responsible for managing the indexing of student works and executing searchable encryption requests;
5. IPFS: Responsible for decentralized storage of encrypted student works. IPFS generates corresponding hash values (CID) based on the ciphertext content and then returns the CID to the administrator.

5.2. Searchable Encryption

This paper designs a searchable encryption scheme to protect student works, ensuring their privacy while enabling sharing.

The proposed searchable encryption scheme in this paper consists of six specific algorithms: the key generation algorithm $KeyGen$, index generation algorithm $IndexGen$, encryption algorithm $Encrypt$, trapdoor generation algorithm $TrapGen$, search algorithm $Search$, and decryption algorithm $Decrypt$. The description is as follows:

(1) $KeyGen(\lambda) \rightarrow (SK)$. This function initializes the key. The administrator provides the security parameter λ, and through this algorithm, the AES-CBC-256 encryption algorithm's encryption and decryption key SK can be output. The authorized user receives the key SK for search via a secure channel;

(2) $IndexGen(F, W, Hash) \rightarrow I_H$. First, the administrator selects a collection of student works $F = \{F_1, F_2 \ldots, F_N\}$. The set of keywords composed of files F is $W = \{W_1, W_2 \ldots, W_N\}$. The SHA-256 hashing algorithm hashes each keyword and then stores the hash codes on the blockchain;

(3) $Encrypt(F, SK) \rightarrow (CT)$. The administrator encrypts the student's work $F = \{F_1, F_2 \ldots, F_N\}$ using the AES-CBC-256 encryption key SK. The administrator then uploads the encrypted works to IPFS for storage. The blockchain records the hash CID (Content Identifier) and other work information that IPFS returns. The corresponding work information binds each encrypted work's CID;

(4) $TrapGen(Q, hash) \rightarrow T_H$. The SHA-256 hashing algorithm generates the hash value TH of a search keyword Q when a student or teacher inputs it. We then upload this trapdoor TH to the blockchain for querying operations;

(5) $Search(I_H, T_H) \rightarrow CID$. Upon receiving the query trapdoor T_H from the user, the blockchain performs a precise search operation based on the index I_H deposited by the administrator. Ultimately, the blockchain returns the CID information of the successfully found works to the user;

(6) $Decrypt(CT, SK) \rightarrow F$. The user first receives the search result CID returned from the blockchain. Then, they use the hash value CID to download the file from IPFS. Finally, using the decryption key SK obtained from a secure channel, they can decrypt and obtain the desired work's plaintext.

5.3. Role-Based Access Control

This section primarily discusses access control in relation to sharing works. The entity model introduction and the system implementation section in the next section contain other recognized permissions. Teachers have access to their students' work. If they need to access the works of students taught by other teachers, they must request permission from the administrator. Students can access the works of other students within their class. If they need access to the work of students from different classes or previous years, they must also request permission from the administrator. The administrator only grants access to requested content after receiving approval.

5.4. Smart Contract Design

In the proposed scheme, the smart contracts involved are described as shown in Table 3.

Table 3. Description of smart contracts.

Smart Contract	Main Contract	Description
User Contract	Administrator Contract	Store administrator's login account and hashed password
	Student Contract	Chain student information: upload, update, delete, query student information
	Teacher Contract	Chain teacher information: upload, update, delete, query teacher information
Course Contract	Course Contract	Chain course information: upload, update, delete, query course information
	Work Contract	Chain work information: upload, update, delete, query work information
Score Contract	Regular Conditions Contract	Chain conditions for full marks in weekly regular grades: upload, update, query full marks conditions
	Regular Scores Contract	Chain weekly regular grades: upload, update, query regular grades
	Average of Regular Scores Contract	Calculate the average of weekly regular grades summed across all weeks, update the average of regular grades, query the average of regular grades, and retrieve rankings based on regular grades.
	Midterm Scores Contract	Chain midterm exam grades: upload, update, query midterm grades, retrieve ranking based on midterm grades
	Final Scores Contract	Chain final exam grades: upload, update, query final exam grades, retrieve ranking based on final exam grades
	Comprehensive Scores Contract	Chain comprehensive grades: upload, update, query comprehensive grades, retrieve ranking based on comprehensive grades
Access Control Contract	Access Require Contract	Process applications from students and teachers (approve/reject)

5.5. Time Analysis of Searchable Encryption

This solution entails downloading 1000 English papers from the internet using a web crawler. We employ the TF-IDF and N-gram algorithms to extract two-word keywords from the titles of these papers. We sort each keyword by its TF-IDF score and select the top n features (where $n = 1$) as the keywords for each paper. Consequently, 2000 keywords (1000 sets of keywords) are chosen for the 1000 papers.

5.5.1. Index Generation Time

As the number of keywords increases, the index generation time exhibits linear growth, as illustrated in Figure 5a. Similarly, with an increase in the number of documents, the index generation time increases linearly, as shown in Figure 5b. The figures demonstrate that this study's indexing time remains within the millisecond range, indicating minimal overhead.

5.5.2. Trapdoor Generation Time

Like the indexing stage, trapdoor generation time increases linearly with an increase in query keywords yet remains within the millisecond range, indicating minimal overhead, as illustrated in Figure 6.

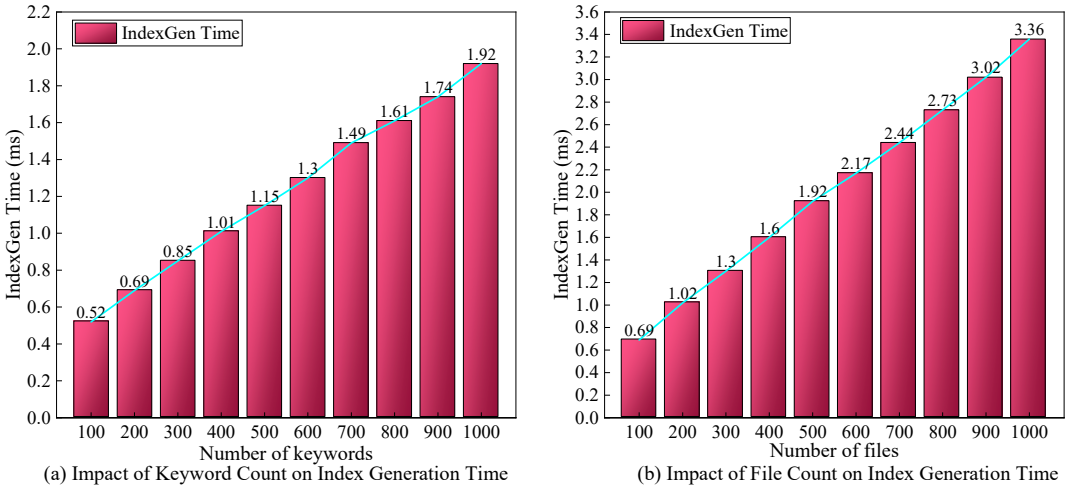


Figure 5. IndeGen time.

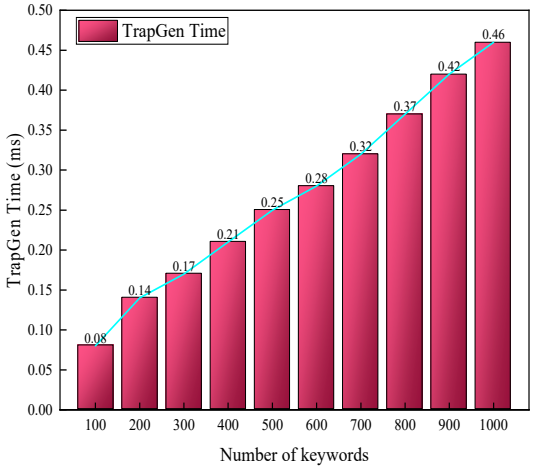


Figure 6. TrapGen time.

5.5.3. Search Time

As shown in Figure 7, the number of files is a crucial factor affecting search time, with search time increasing correspondingly as the number of files increases. In this study, even when processing 1000 files, the search time is only 0.03 ms, demonstrating that the search time is highly efficient and practical.

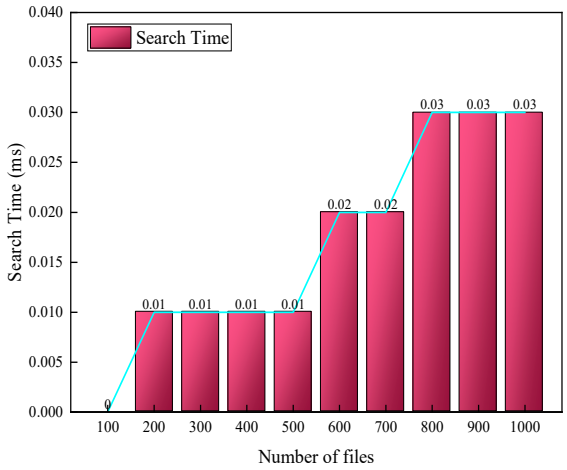


Figure 7. Search time.

6. System Implementation and Deployment

B-Education (Blockchain-Education) is a blockchain-based blended teaching and evaluation system. It integrates the Interplanetary File System to store learning data from multiple teaching platforms, access control, searchable encryption technology to realize authorized searches of encrypted data, and the blockchain platform to ensure the security and integrity of educational data. This section discusses the system's implementation and deployment in detail.

6.1. Experimental Environment

The experiment in this paper is carried out on the Windows 10 operating system, using the Truffle framework and Ganache to build the Ethereum blockchain development environment, as detailed in Table 4, and using Nodejs for programming. The hardware environment is as follows: The desktop computer's CPU is a 12th Gen Inter(R) Core(TM) i7-12700 @ 2.10 GHz (Intel Corporation, Santa Clara, CA, USA), and the memory size is 16 G. The software environment is Ganache 7.9.1, Truffle 5.11.5, Solidity 0.5.16, Web3 1.10.0, Go-IPFS 0.7.0, Remix 1.3.6, VSCode 1.90.0, and Nodejs 20.13.1.

Table 4. Development environment.

Environment	Name	Version
hardware environment	CPU	12th Gen Inter(R) Core(TM) i7-12700 @ 2.10 GHz
	RAM	16 G
software environment	Ganache	7.9.1
	Truffle	5.11.5
	Solidity	0.5.16
	web3	1.10.0
	go-ipfs	0.7.0
	Nodejs	20.13.1
	Remix	1.3.6
	VSCode	1.90.0

6.2. System Overall Design

The overall design framework of the system is the skeleton of the system, supporting the operation of the entire system. System design is the foundation for detailed design and can guide subsequent development work. We have developed a teaching management platform suitable for higher education systems based on specific needs and management processes. We developed the system using high-level development languages and adopted a browser/server (B/S) architecture to provide higher-level teaching management support for the field of higher education [48,49]. The system includes basic modules like information management, teacher management, student management, grade management, and work management, as well as functions like adding, deleting, querying, and modifying. The blockchain-based multivariate blended teaching evaluation system uses Ganache as the test blockchain, combines the IPFS to implement file storage, and uses Solidity and Node.js languages for smart contract and server development, as shown in Figure 8 for details.

6.3. System Operation

Figure 9 below shows a schematic diagram of the blockchain teaching and evaluation system, describing the roles in the B-Education system, their functions, operational privileges, and the steps for using the system.

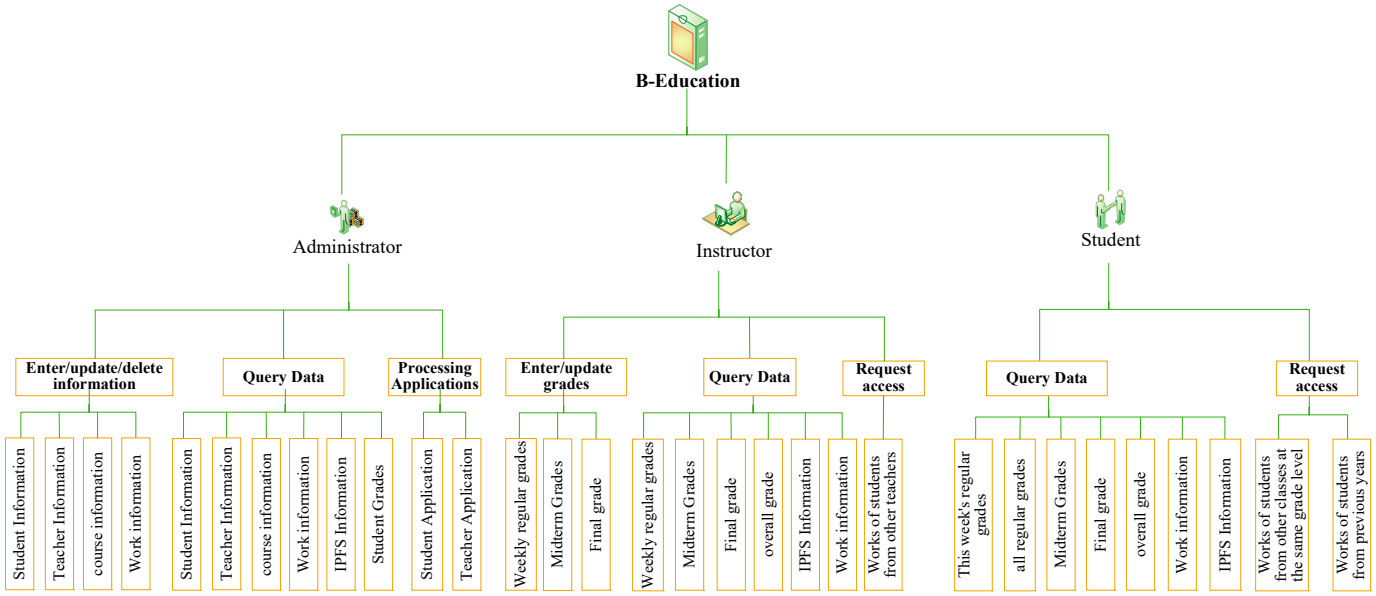


Figure 8. System overall design.

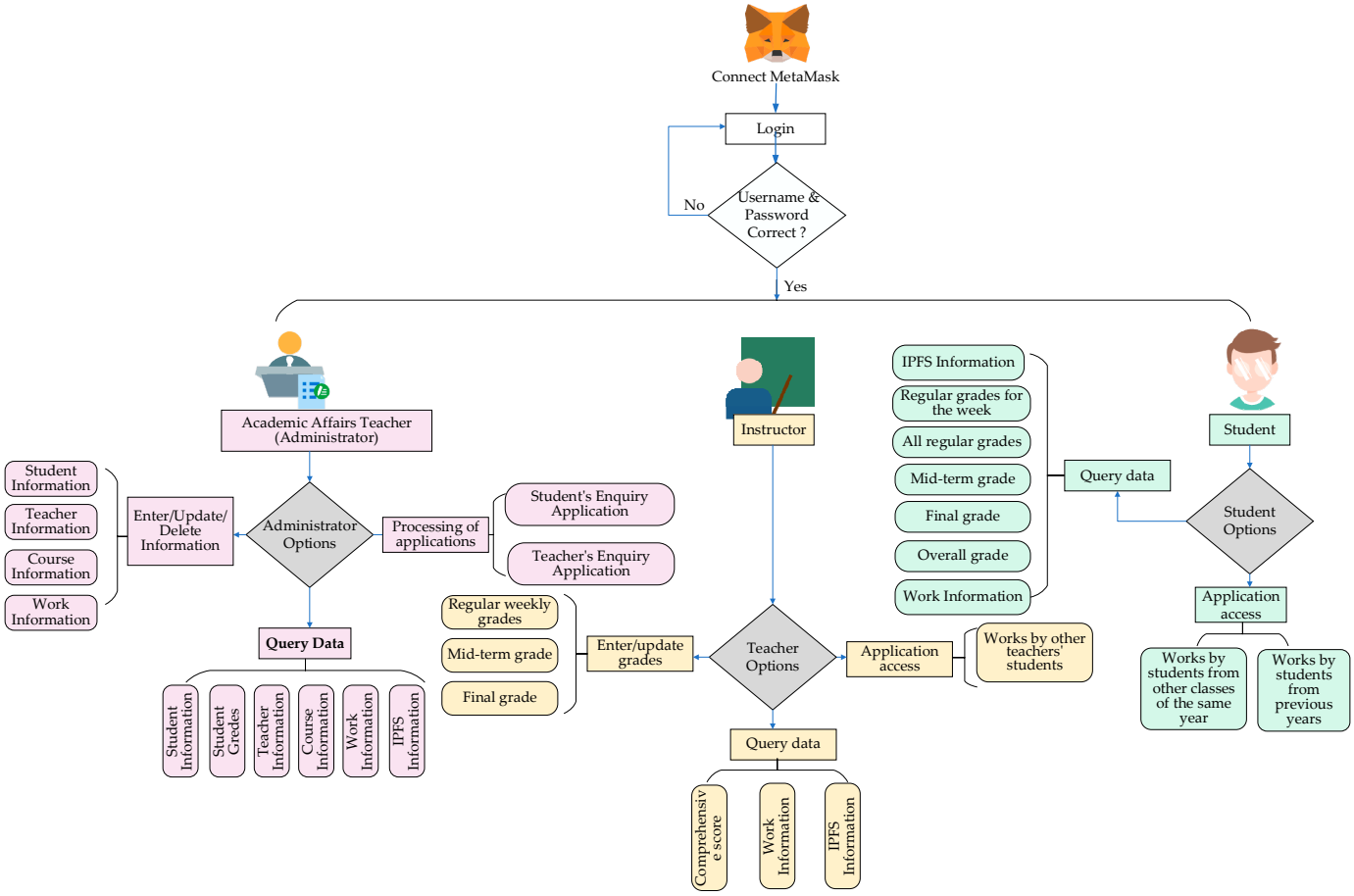


Figure 9. System diagram.

To start using the system, users must establish a connection with the MetaMask wallet and log in securely using a unique username and password. After login verification, the system will identify the user's role, which may be that of an academic affairs teacher (administrator), a course teacher, or a student. The corresponding navigation bar will direct the user to perform specific functions depending on the role.

Administrators can enter or update student, teacher, and course information through the navigation bar. They can also query student information, teacher information, course information, student work, and student grades. They can also delete student information, teacher information, course information, and student work, but they cannot delete or modify student grades. Additionally, they can process student and teacher application access requests.

Teachers can input students' weekly regular grades, midterm grades, and final grades and query students' overall grades, works, and IPFS information. Teachers must apply for administrator access to inquire about the work of other teachers' students.

Students can check this week's regular grades, all weeks' regular grades, midterm grades, final grades, comprehensive grades, work information, and work IPFS information. If students need to check the work information of other classes in the same year and the work information of students from previous years, they need to apply for access rights from the administrator.

Administrators, teachers, and students in the system use precise searchable encryption technology when searching for works to ensure data security and privacy protection.

6.4. Main Pages of the System

Due to space constraints, we have selected some screenshots of the critical page.

6.4.1. Administrator Process Student Applications Page

When students need to apply to view the works of senior students from previous years or those from other classes in the same grade, they must submit a request to the administrator. Access will only be granted once the administrator approves the request. This functionality and processing is similar to the teacher application feature, as shown in Figure 10.

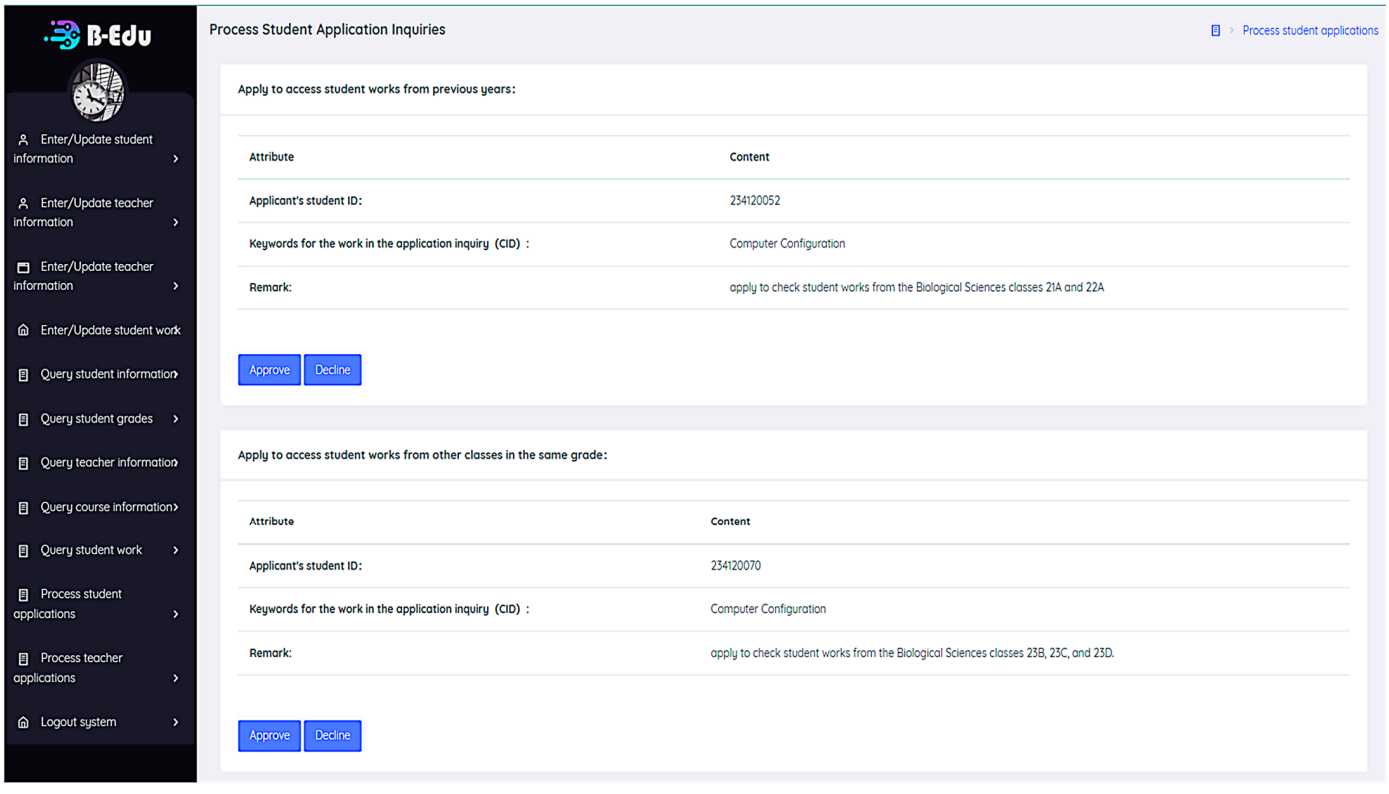


Figure 10. Processing application.

6.4.2. Teachers Enter Grades

Figure 11 illustrates how teachers enter their students' regular grades. The teacher first enters the conditions for the student's regular grades to be full marks this week: the number of attendances, online self-study times, and classroom discussions. The teacher then enters the students' actual attendance times, online self-study times, classroom discussions, and comprehensive experimental design and homework scores this week.

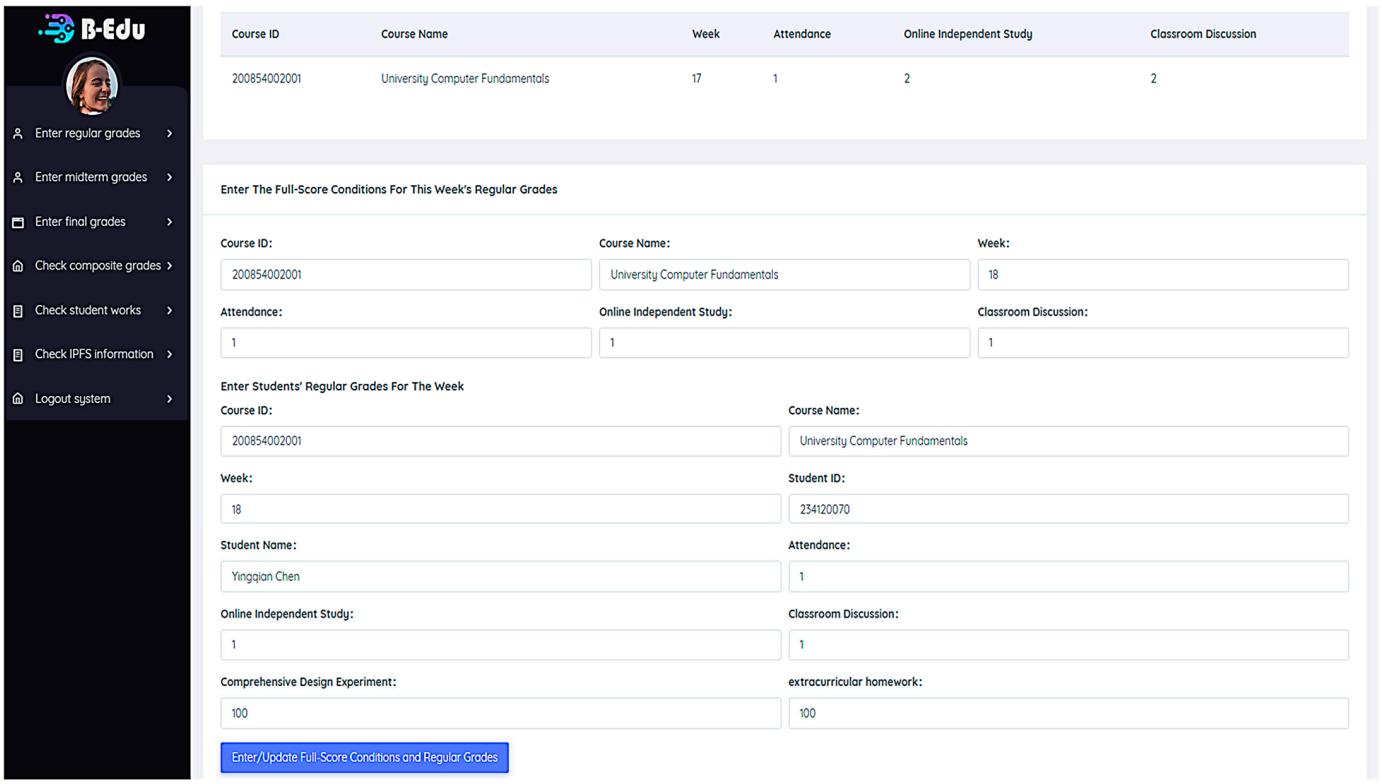


Figure 11. Entering regular grades.

Figure 12 shows the teacher entering the student's final grade. The teacher first searches for the student using student ID, then inputs course ID, course name, student number, and student name. Finally, they enter the student's subjective and objective grades. The final grade is composed of subjective and objective grades.

6.4.3. Student Inquire All Weekly Grades

Students can query their course's detailed weekly performance grades by entering the course number (Figure 13). Administrators and teachers can also query a student's detailed performance grades by entering both the course and student ID numbers.

6.4.4. Students Inquire Comprehensive Scores and Rankings

Figure 14: Students can input the course ID to query their regular grades and rankings, midterm grades and rankings, final grades and rankings, and comprehensive grades and rankings. Similarly, administrators and teachers need to enter the course ID and student ID to query the details of the student's comprehensive grades.

6.4.5. Student Search Works

Figure 15: Students can input their student ID to search for specific students' works or enter keywords to search for all students' works that meet the criteria. If a student wants to

search for student works from previous years or other classes of the same grade, they must apply to the administrator for access.

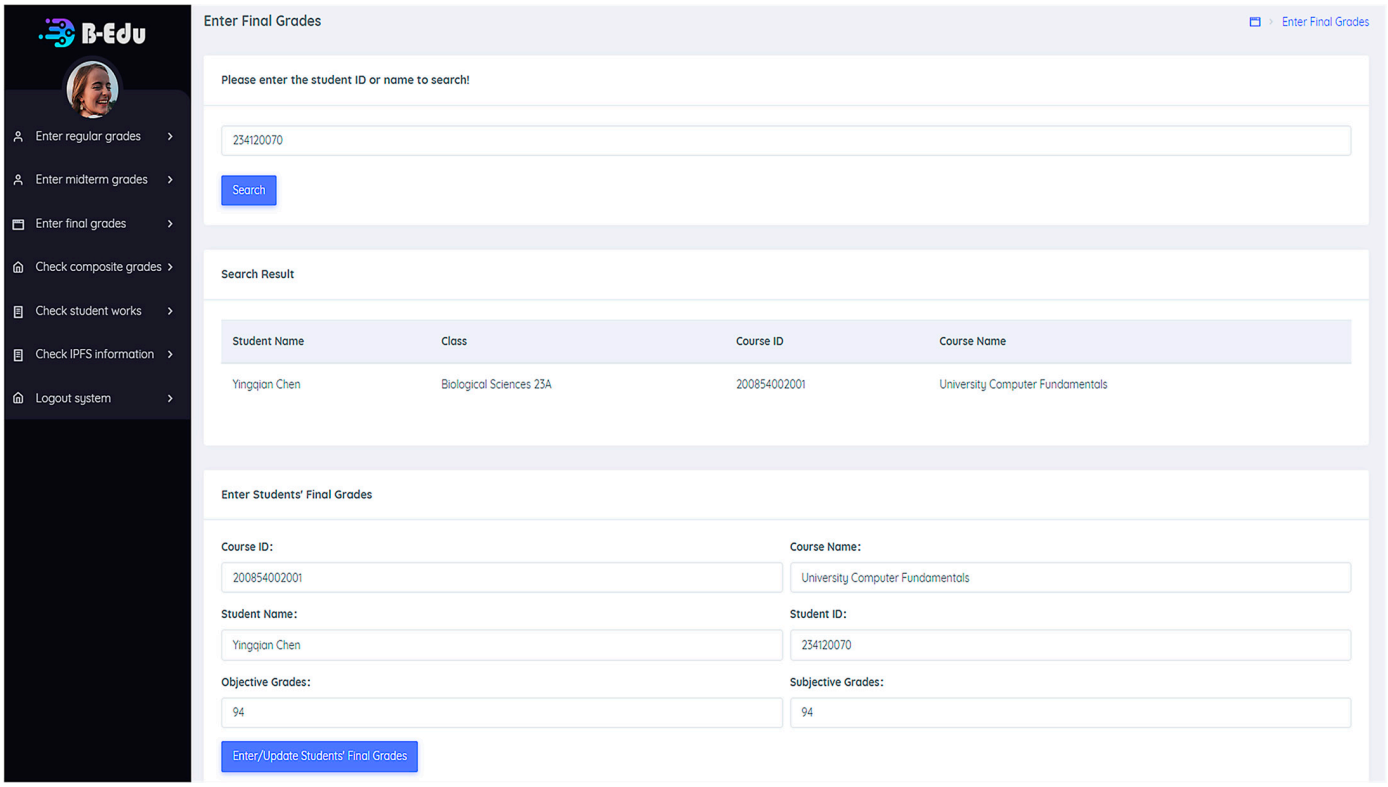


Figure 12. Entering final grades.

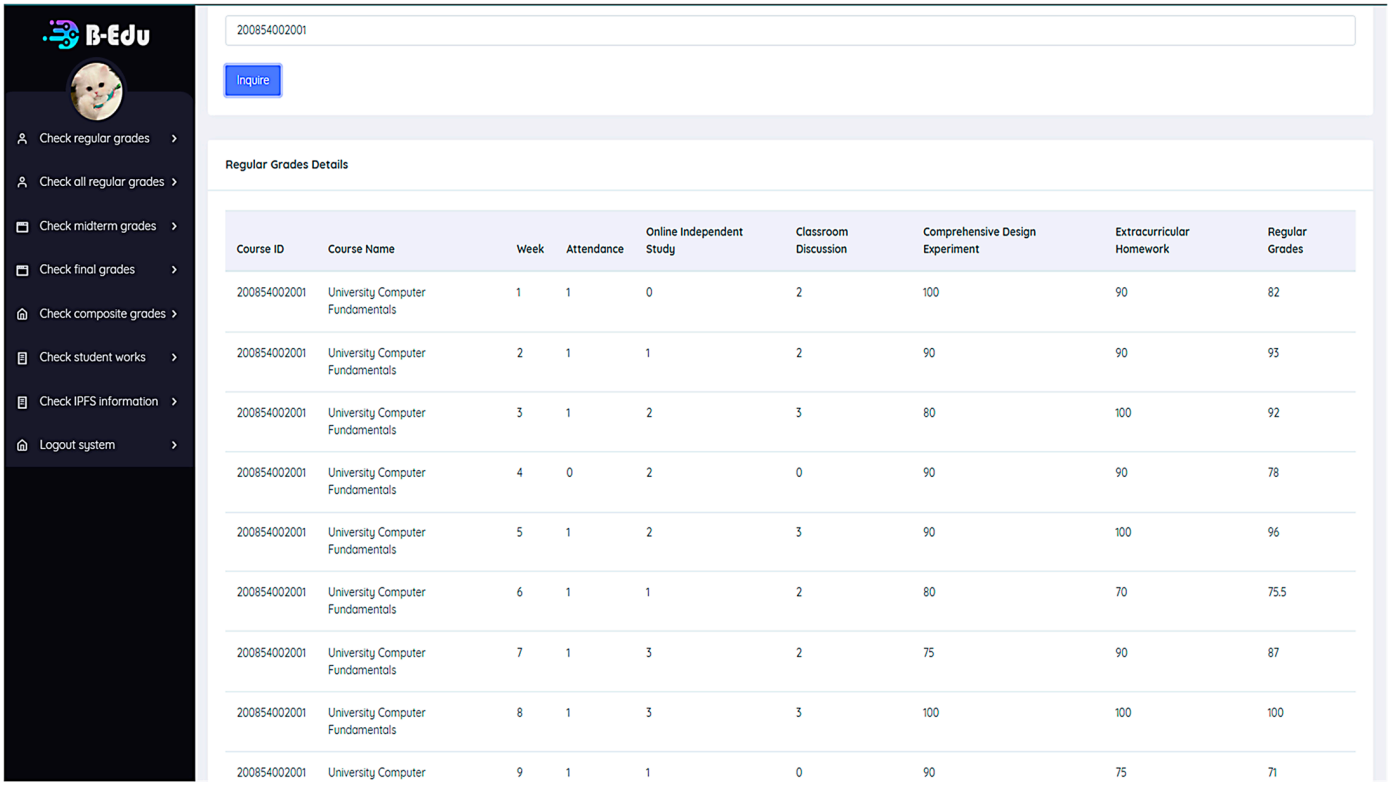


Figure 13. Querying regular grades.

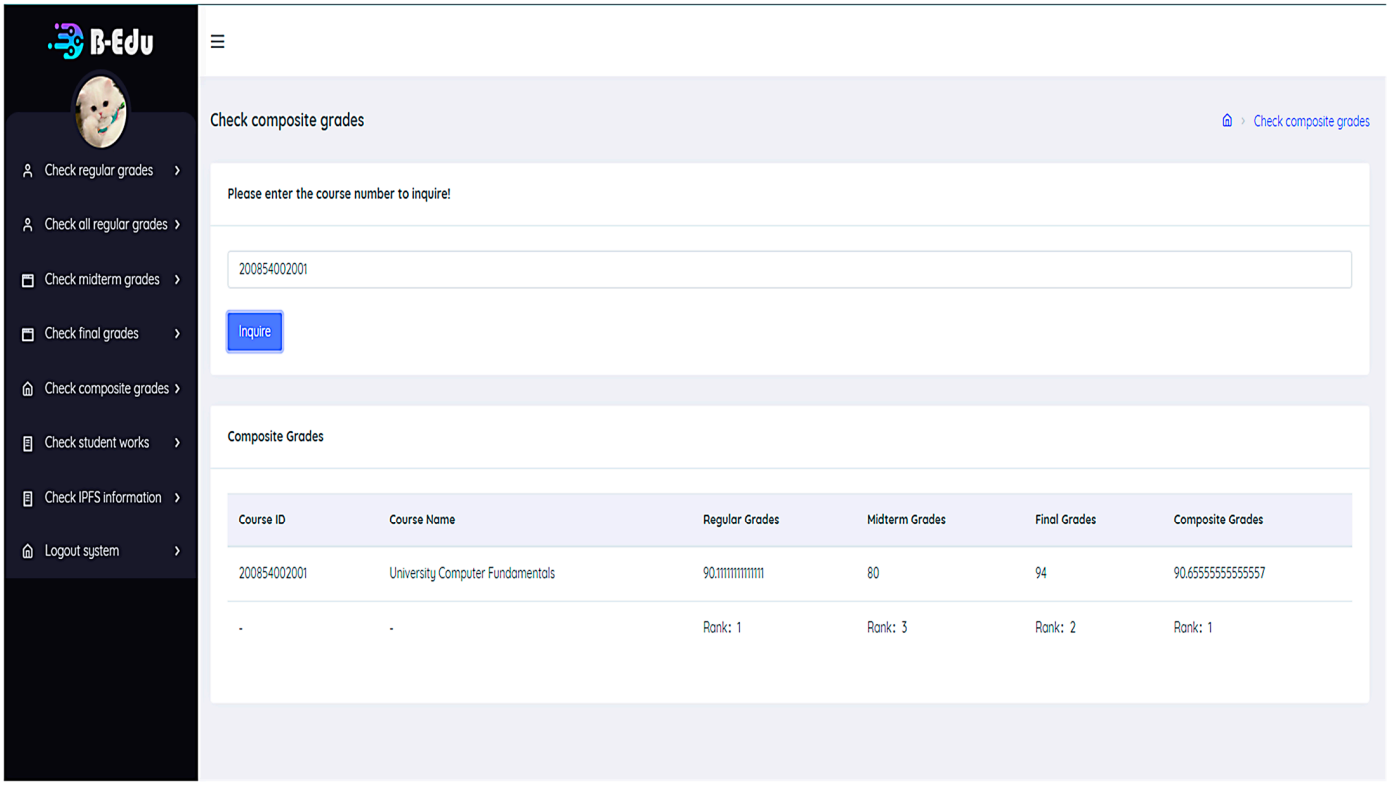


Figure 14. Querying comprehensive grades and ranking.

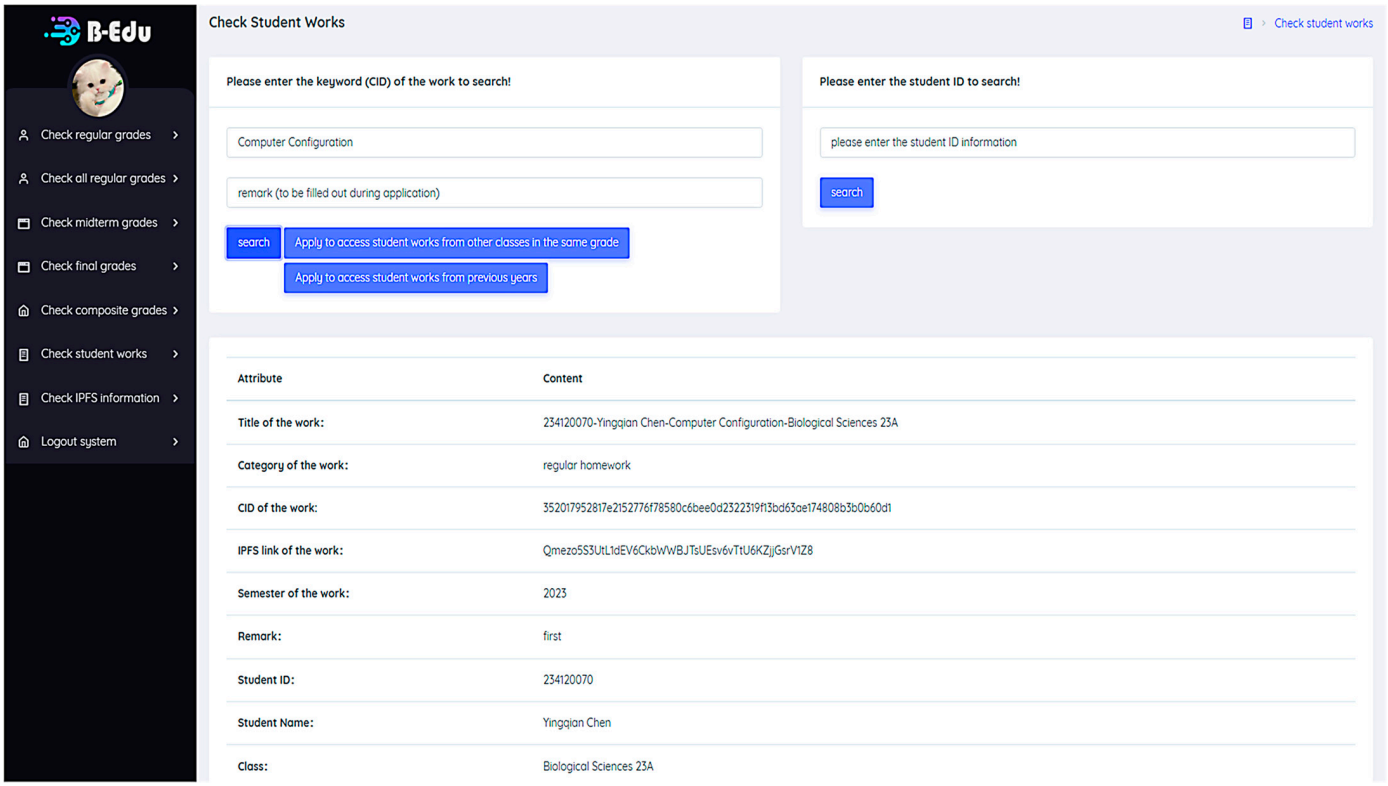


Figure 15. Searching for works.

6.5. System Gas Consumption Analysis

In the Ethernet Ganache test network, the gas price value is 20,000,000,000 (GWei), and the gas limit value is 6,721,975 (GWei), as shown in Figure 16. The transaction cost formula for Ethernet and the maximum transaction cost formula are shown below.

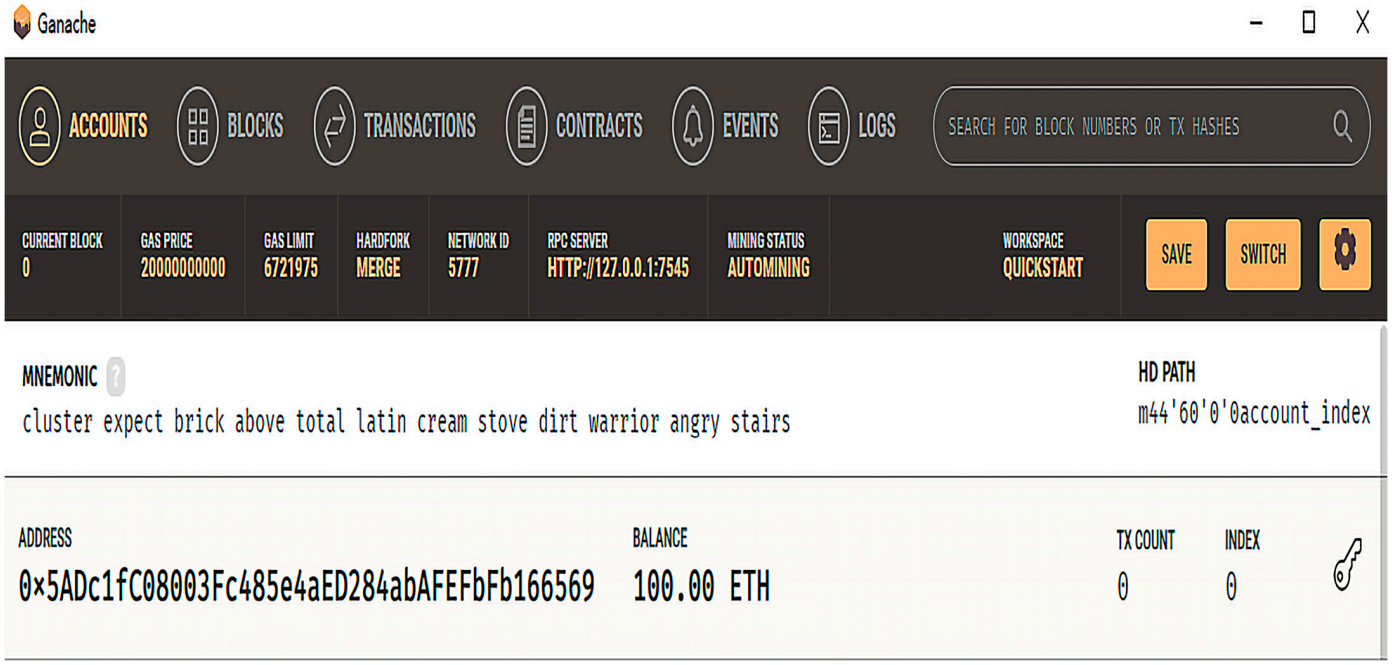


Figure 16. Ganache interface.

$$\text{Gas Used Fee} = \text{Gas Used} \times \text{Gas Price}$$

$$\text{Gas Limit Fee} = \text{Gas Limit} \times \text{Gas Price}$$

In Ethereum smart contract transactions, the first transaction's gas consumption is frequently the highest. For administrators, teachers, and students, we take the gas value of the first transaction as a reference for the gas consumption of each identical transaction operation. This is because the first transaction will interact with the smart contract, load the contract code, initialize some state variables or store data. The Ethereum Virtual Machine (EVM) and the underlying storage system need to perform additional initialization work, resulting in additional computing overhead and consuming more gas.

Subsequent identical transaction operations must only call the deployed contract method, and the initialized data must be read and updated. Furthermore, subsequent operations can utilize Ganache's internal caching mechanism—factors that significantly reduce gas consumption.

In summary, the gas value of the first transaction operation is the highest. As long as the first gas value does not exceed the gas limit, the gas consumption of subsequent transaction operations of the same type will not exceed the first gas value or the gas limit.

6.5.1. Smart Contract Deployment Gas Consumption

Figure 17a shows the gas consumption when deploying the smart contract on the chain. The smart contract occupies seven blocks, and the gas consumption of each block does not exceed the gas limit. The contract transaction can be successfully executed. Figure 17b shows the transaction fees for deploying smart contracts on the chain. The Ether spent on the transaction did not exceed the limit on transaction fees.

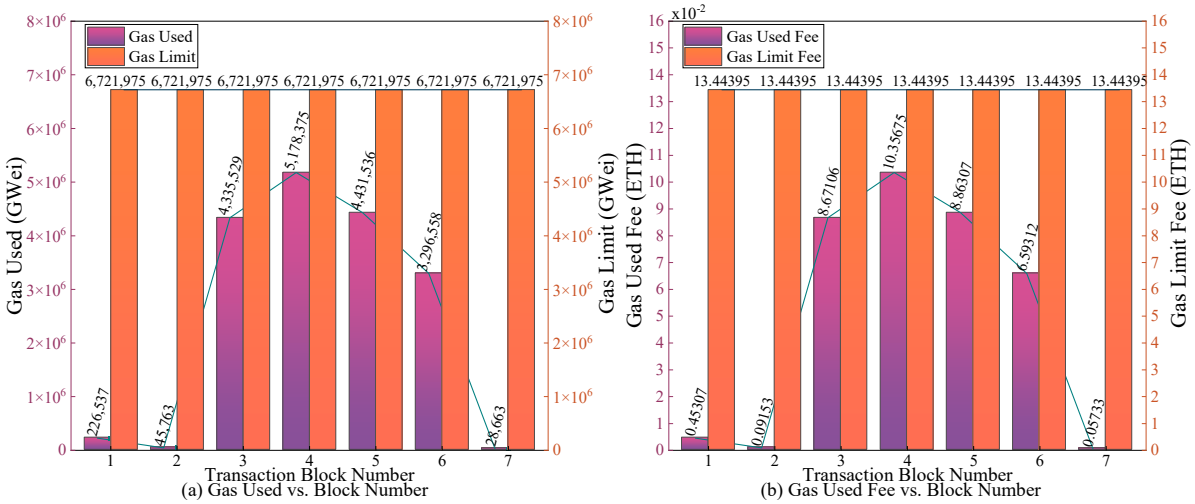


Figure 17. Smart contract deployment gas cost.

6.5.2. Administrator Gas Consumption

Figure 18 shows the administrator's gas consumption during certain transaction operations. The transaction operations include adding/modifying/deleting student information, teacher information, course information, student work information, and processing teacher and teacher applications. Processing application 1 is to review student applications that students want to inquire about students' work from previous years. Processing application 2 is to review students' applications that students want to query the works of students in other classes of the same grade. Processing application 3 is to review teachers' applications that teachers want to query the works of other teachers' students. As shown in the figure, each administrator transaction did not exceed the gas limit, and all transactions were executed and completed successfully.

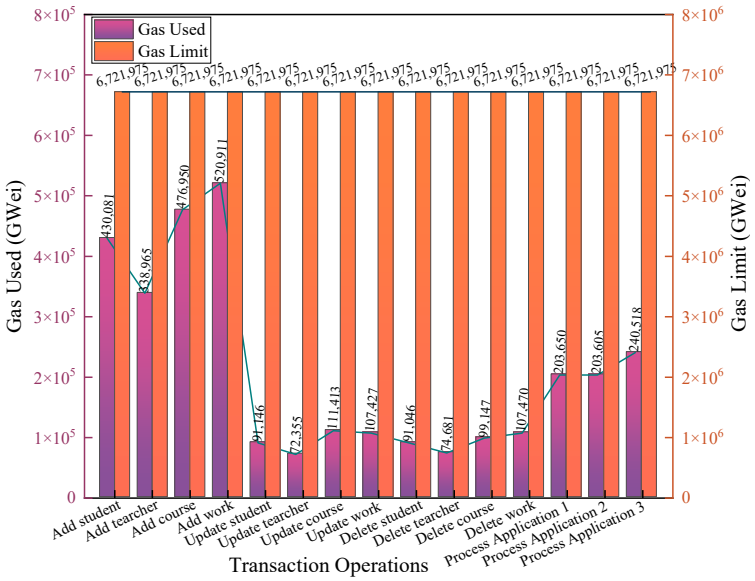


Figure 18. The gas cost of administrator transaction operations.

Figure 19 shows the Ether transaction fees paid by the administrator during each transaction operation. The figure shows that each transaction's Ether expenditure did not surpass the fee limit.

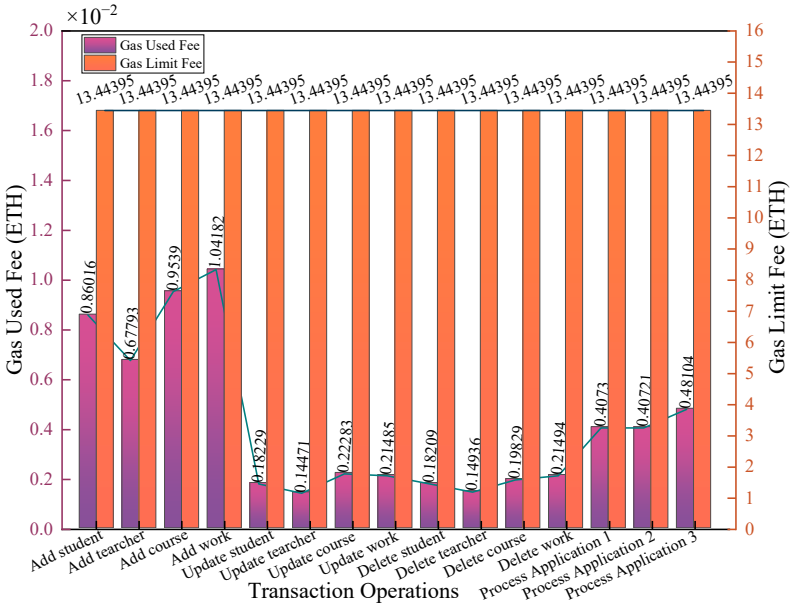


Figure 19. ETH cost of administrator transaction operations.

6.5.3. Teacher Gas Consumption

Figure 20 shows teachers' gas consumption during certain transaction operations. The transaction operations include logging into the system, adding/modifying the full score conditions for weekly regular grades, adding/modifying the weekly regular grades, midterm grades, final grades, comprehensive grades, the average of regular grades for all weeks of the semester, and applying to access other teachers' student works. As shown in the figure, each teacher transaction did not exceed the gas limit, and all transactions were executed and completed successfully.

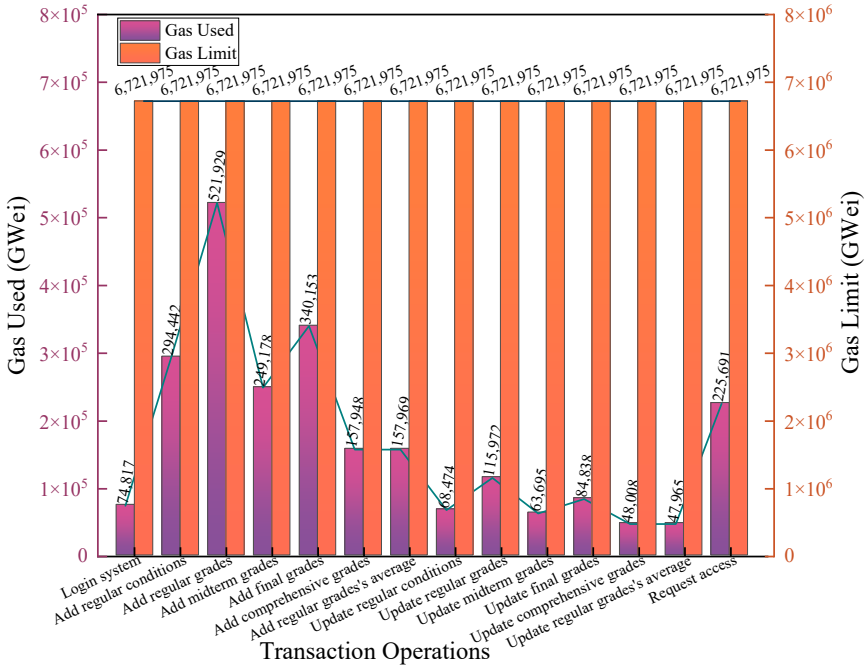


Figure 20. The gas cost of teacher transaction operations.

Figure 21 shows the Ether transaction fees paid by the teacher during each transaction operation. The figure shows that each transaction's Ether expenditure did not surpass the fee limit.

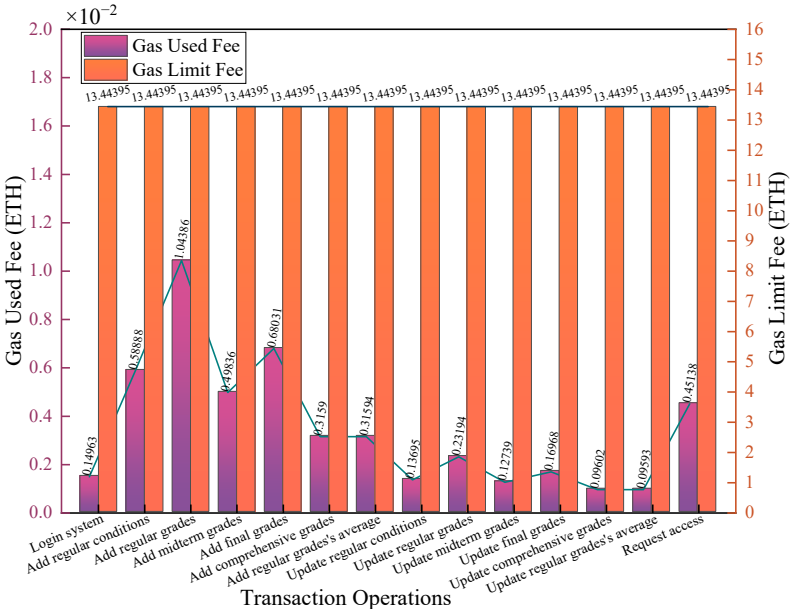


Figure 21. ETH cost of teacher transaction operations.

6.5.4. Student Gas Consumption

Figure 22a shows students' gas consumption during certain transaction operations. The transaction operations include logging into the system and applying for access control. Application 1 is for the students who want to inquire about students' work from previous years. Application 2 is for students who want to query students' works in other classes of the same grade. As shown in the figure, each student transaction did not exceed the gas limit, and all transactions were executed and completed successfully. Figure 22b shows the Ether transaction fees paid by the teacher during each transaction operation. The figure shows that each transaction's Ether expenditure did not surpass the fee limit.

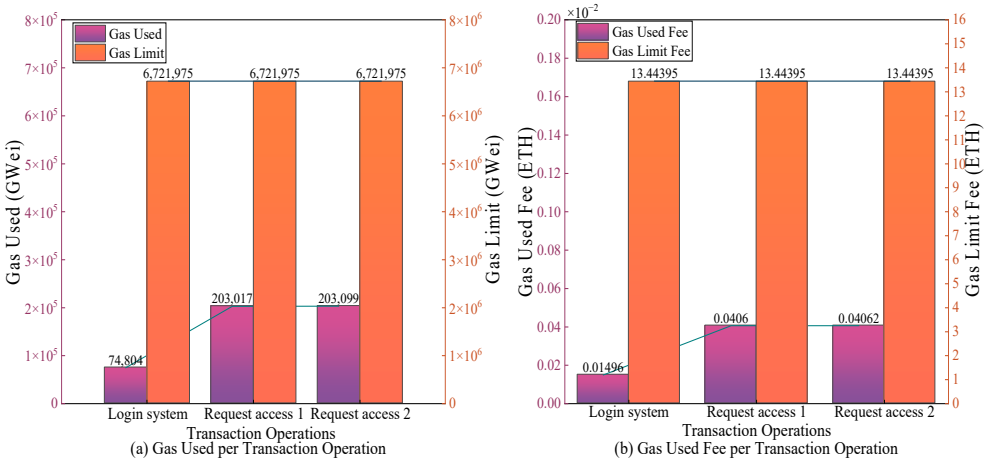


Figure 22. The gas cost of student transaction operations.

In general, whether it is the deployment of smart contracts or each transaction of administrators, teachers, and users, the gas consumption has not exceeded the gas limit, and the expenditure of Ether has not exceeded the gas fee limit. This implies that the blockchain system we developed can successfully execute all transactions and contract deployment calls, and there will be no contract execution failure or transaction cancellation due to exceeding the gas limit. The developed system is also feasible if deployed and used in actual scenarios.

6.6. Empirical Results

Based on the developed system, we encrypted the works of 100 students and stored them in IPFS, while the other information related to the works was stored on the blockchain. Additionally, we recorded each student's weekly grades, midterm grades, and final grades. Through a series of manual tests on the system's functionalities, we confirmed that all functionalities were successfully implemented. Whether adding, deleting, or modifying information, providing a search function that meets access permissions, or calculating the overall grade based on different grade weights, the system operates correctly. Furthermore, the rankings for regular, midterm, final grades, and comprehensive grades are all correctly displayed, demonstrating that the system meets practical application requirements.

6.7. Scalability Analysis

This system uses IPFS to store AES-CBC-256 encrypted ciphertext files. To ensure the privacy of student works, universities can opt to use their own distributed cloud servers for storing these works instead of relying on IPFS. Currently, student works are encrypted using AES-CBC-256, but administrators can choose to employ other encryption algorithms, such as RSA or ECC, provided that key management is appropriately handled. The system's searchable encryption functionality currently supports only exact keyword searches; however, it can be expanded in the future to support fuzzy searches, allowing users to search for keywords with a certain degree of error, thereby enhancing the user experience. In terms of security, there are plans to replace SHA-256 with the HMAC function to further strengthen the security of the index and trapdoors. Additionally, techniques such as digital signatures and hashing can be introduced to verify the correctness and integrity of search results, ensuring data accuracy and completeness. While the current system employs role-based access control (RBAC) for access control, future developments may introduce Attribute-Based Access Control (ABAC) to enable more granular access management, supporting more complex and flexible access control strategies. Additionally, any university can adopt its own teaching evaluation model tailored to its specific needs. By simply adjusting the grade weights through the smart contract, the system can be flexibly adapted to meet the requirements of other universities.

7. Security Analysis and Formal Verification

7.1. Security Analysis

7.1.1. Confidentiality of Shared Data

In this paper's solution, administrators upload student works to IPFS after AES-CBC-256 encryption, and the blockchain stores the hash value returned by IPFS. The security of the AES-CBC-256 algorithm has been demonstrated in other papers, so the privacy of students' work is better guaranteed. Because the AES-CBC-256 algorithm has the same encryption key and decryption key, administrators must strictly protect the key. This paper assumes that the administrator sends the key to the authorized user through a secure channel. We do not currently discuss the implementation of the secure channel. Because the administrator randomly generates the key and remains unknown to the blockchain or other individuals, the document's safety is ensured. It will not lead to the leakage of student work, thereby safeguarding the confidentiality of the shared data associated with student work.

Additionally, the administrator sets access control for student work data and assigns access privileges based on the user's role to protect the privacy of student work data. Teachers and students conduct searches of student work data under established access controls, ensuring student privacy and preventing unauthorized search access.

7.1.2. Secure Distributed Storage of Encrypted Data On-Chain and Off-Chain

The scheme in this paper uploads the student work files into IPFS after AES-CBC-256 encryption, storing the corresponding ciphertext CID on the blockchain to lessen the storage pressure on the blockchain. IPFS divides encrypted files and stores them in a distributed manner in the system's storage nodes, making it difficult for attackers to obtain complete encrypted files and decrypt them. We can accurately search the ciphertext CID only when the keyword index and trapdoor on the blockchain match fully. The attacker cannot crack the ciphertext data without the corresponding decryption key, even if they obtain it illegally. Additionally, because IPFS and blockchain are decentralized, there will not be a problem with data loss due to a storage node failure, and intermediaries' security risks will not exist. Meanwhile, IPFS's hash verification mechanism and blockchain's tamper-proof advantage ensure the integrity of the data cipher.

7.1.3. Confidentiality of Indexes and Trapdoors

This paper's scheme converts keywords into file indexes and query trapdoors. We process the keywords using the SHA-256 hash algorithm to generate 256-bit hash codes. This process is irreversible and therefore does not reveal any original information about the keyword. The keyword index stored on the blockchain is a processed hash code, and the whole search process is carried out based on the verification of matches between hashes. The blockchain cannot obtain information about the original keywords or ciphertext data. This means that the blockchain cannot infer the accurate information of any document or keyword from indexing and trapdoors, thus ensuring the privacy and security of indexing and query trapdoors.

7.1.4. Potential Attack Vectors and Mitigation Strategies

The system faces various potential security threats, including data poisoning, privacy leakage, DDoS attacks, and unauthorized access. To effectively address these risks, the system integrates blockchain, IPFS, searchable encryption, and role-based access control (RBAC) technologies to construct a multi-layered defense architecture. Specifically, the immutability of blockchain prevents data poisoning by ensuring that data cannot be tampered with during storage and transmission. Leveraging IPFS's content addressing mechanism, the system can withstand DDoS attacks and avoid single points of failure causing system downtime. In terms of privacy protection, the system uses searchable encryption to ensure the confidentiality of data during transmission and storage, enabling secure retrieval without disclosing the data content. For unauthorized access, the system employs role-based access control, enforcing fine-grained access policies to ensure that only authorized users can access specific data. These security mechanisms work in tandem to provide robust security for the system, significantly enhancing its resilience and ability to withstand attacks.

7.2. Formal Verification

Formal methods are essential for verifying the correctness of smart contract code, effectively preventing costly errors and security vulnerabilities. This paper utilizes Slither [50], a leading static analysis tool, for formal verification. As illustrated in Figure 23, the process begins with the Solidity compiler compiling the contract source code into an Abstract Syntax Tree (AST), which serves as input for Slither. In the first stage, Slither retrieves critical information, generating inheritance graphs, control flow graphs (CFGs), and expression lists. Next, Slither converts the code into its intermediate representation language, SlithIR, which leverages the Static Single Assignment (SSA) form to optimize code analysis. Finally, Slither runs predefined analysis modules to provide valuable insights into other compo-

nents, such as data dependencies, variable read/write operations, and protected function calls. Through Slither's core processing, it ultimately achieves vulnerability detection, code optimization suggestions, and code comprehension outputs, significantly enhancing the security and performance of smart contracts.

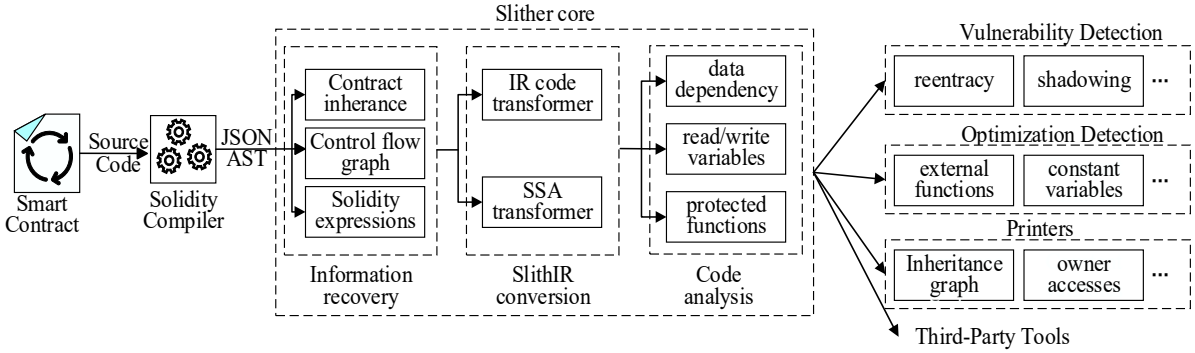


Figure 23. Slither overview.

As shown in Table 5, compared to other formal verification tools such as Securify, SmartCheck, and Solhint, Slither demonstrates superior performance in terms of accuracy, efficiency, and robustness. Slither not only exhibits the lowest false positive rate and can correctly parse 99.9% of all public Solidity code, but it is also the only tool capable of identifying two real-world reentrancy vulnerabilities. These advantages are the key reasons why we selected Slither for formal verification.

Table 5. Summary of the evaluation results.

		Slither	**Securify**	**SmartCheck**	**Solhint**
Accuracy	False positives	10.9%	25%	73.6%	91.3%
	Flagged contracts	112	8	793	81
	Detections per contract	3.17	2.12	10.22	2.16
Performance	Average execution time	0.79 ± 1	41.4 ± 46.3	10.9 ± 7.14	0.95 ± 0.35
	Timed out analyses	0%	20.4%	4%	0%
Robustness	Failed analyses	0.1%	11.2%	10.22%	1.2%
Reentrancy examples	DAO	√	×	√	×
	Spankchain	√	×	×	×

In the table, the symbol '×' denotes that the tool is not resistant to reentrancy attacks, whereas '√' denotes that it is resistant.

We selected the latest version of Slither, version 0.10.4, which includes 93 detectors. Slither categorizes the severity of issues using color codes: red indicates high-risk, critical errors or vulnerabilities that must be addressed, such as reentrancy attacks, integer overflows, or unsafe external calls; yellow signifies medium-level warnings that, while not fatal, still require attention, such as excessive gas consumption or unhandled return values; and green represents code specification suggestions, which generally do not impact security or functionality and can be addressed at the developer's discretion. When addressing Slither's findings, red and yellow issues should be prioritized to ensure the security and reliability of the contract, followed by the consideration of green suggestions to enhance code quality.

As shown in Figure 24, the analysis of the Administrator contract revealed six issues detected by the 93 detectors, all of which are green code specification suggestions. No severe errors (red) or warnings (yellow) were identified. The results indicate that the solc-0.5.0 version is outdated, with a recommendation to upgrade to at least version 0.8.0. The latest version of solc is 0.8.26. To address this issue, the specified version of solc can

be installed and used by running the commands "solc-select install 0.8.0" and "solc-select use 0.8.0". Additionally, the Solidity version declaration in the contract should be updated from "pragma solidity ^ 0.5.0" to "pragma solidity ^ 0.8.0".

```
E:\Desktop\contracts\contract>slither administrator.sol
'solc --version' running
'solc administrator.sol --combined-json abi,ast,bin,bin-runtime,srcmap,srcmap-runtime,userdoc,devdoc,hashes,compact-
INFO:Detectors:
Version constraint ^0.5.0 contains known severe issues (https://solidity.readthedocs.io/en/latest/bugs.html)
        - DirtyBytesArrayToStorage
        - ABIDecodeTwoDimensionalArrayMemory
        - KeccakCaching
        - EmptyByteArrayCopy
        - DynamicArrayCleanup
        - ImplicitConstructorCallvalueCheck
        - TupleAssignmentMultiStackSlotComponents
        - MemoryArrayCreationOverflow
        - privateCanBeOverridden
        - SignedArrayStorageCopy
        - ABIEncoderV2StorageArrayWithMultiSlotElement
        - DynamicConstructorArgumentsClippedABIV2
        - UninitializedFunctionPointerInConstructor
        - IncorrectEventSignatureInLibraries
        - ABIEncoderV2PackedStorage.
It is used by:
        - ^0.5.0 (administrator.sol#1)
solc-0.5.0 is an outdated solc version. Use a more recent version (at least 0.8.0), if possible.
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Contract administrator (administrator.sol#2-22) is not in CapWords
Variable administrator.administrator_id (administrator.sol#4) is not in mixedCase
Variable administrator.administrator_passwd (administrator.sol#5) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:administrator.sol analyzed (1 contracts with 93 detectors), 5 result(s) found
```

Figure 24. Analysis results of the administrator contract.

The analysis also pointed out that the naming conventions do not comply with Solidity's PascalCase (CapWords) and camelCase (mixedCase) standards. According to Solidity conventions, contract names should follow PascalCase, where the initial letter of each word is capitalized, without the use of underscores or spaces; function and variable names should follow camelCase, where the first letter is lowercase and the initial letters of subsequent words are capitalized. To comply with these conventions, the initial letter of the contract name should be capitalized, changing "administrator" to "Administrator", and the variable names "administrator_id" and "administrator_passwd" should be modified to "administratorId" and "administratorPasswd", respectively. These changes will resolve the identified issues.

As shown in Figure 25, the analysis of the Final Scores contract revealed 58 issues detected by the 93 detectors, all of which are green code specification suggestions. No severe errors (red) or warnings (yellow) were identified. The results indicate that the solc version is outdated and that the contract and variable naming conventions do not comply with Solidity's PascalCase (CapWords) and camelCase (mixedCase) standards. These suggestions can be addressed using the same methods outlined for the Administrator contract.

Additionally, the analysis recommends using a cached array length instead of repeatedly referencing the length member of a storage array during each iteration. To improve execution efficiency, a local variable can be defined outside the loop to cache the array length; for example, defining "uint length = qm_c_id.length" outside the "for" loop and modifying the loop condition from "i < qm_c_id.length" to "i < length". This adjustment can reduce the number of storage accesses, thereby enhancing execution efficiency.

The results also suggest that some functions should be declared as "external". Changing "public" function declarations to "external" is recommended because "external" functions can only be called by external contracts or users, while "public" functions can be called both externally and internally. Declaring functions that only require external access as "external" can reduce overhead and improve efficiency.

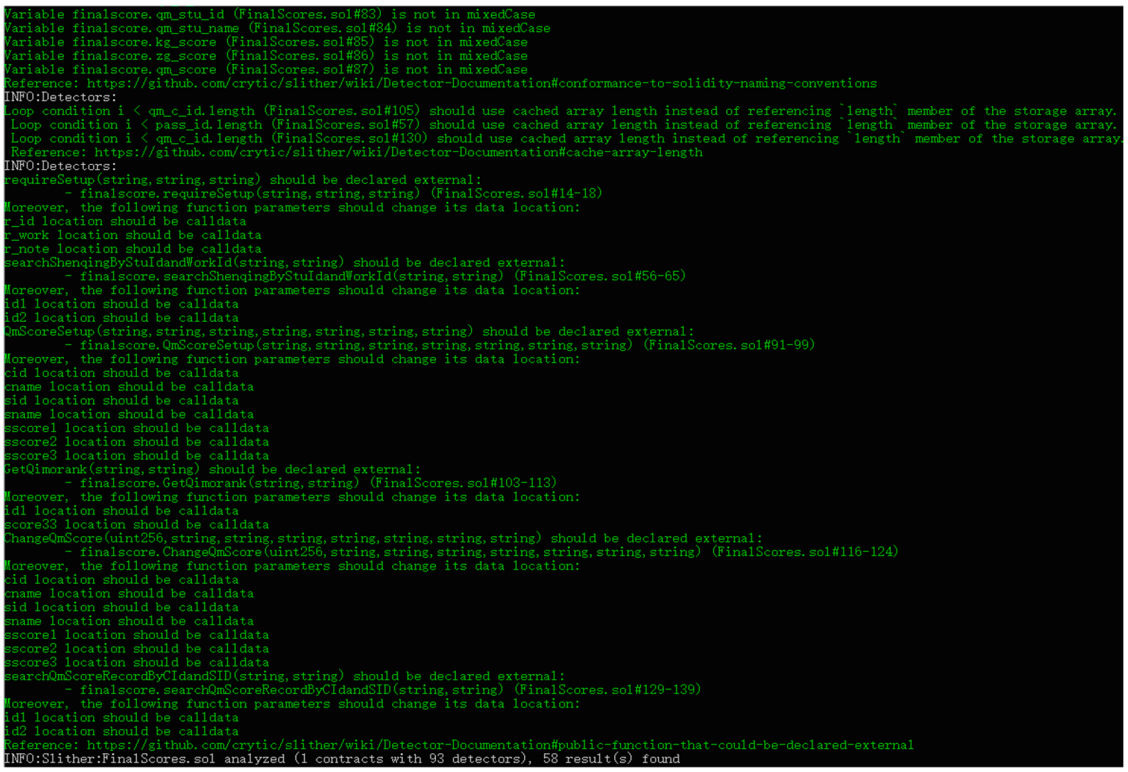


Figure 25. Analysis results of the Final Scores contract.

Another recommendation is to change the data storage location of function parameters from "memory" to "calldata". "Calldata" is a more efficient storage location specifically for function parameters. Since "calldata" is immutable and read-only, it is more gas-efficient than "memory". Using "calldata" can reduce memory copying overhead, further optimizing the contract's execution efficiency.

As shown in Table 6, the analysis results of this contract revealed no severe errors or warnings after a comprehensive static analysis. All findings are related to code specification suggestions. These suggestions primarily aim to enhance the readability and maintainability of the code, rather than addressing functional or security issues. Although these recommendations do not directly impact the security or functionality of the contract, implementing them can improve code comprehensibility, debuggability, and scalability, aligning with higher programming standards. We will appropriately optimize the code based on these suggestions to further enhance the quality and reliability of the contract.

After summarizing the analysis results of all contracts, we identified six main categories of issues, as shown in Table 7. In response to these six categories, we made targeted modifications and optimizations based on the recommendations of the formal analysis tools. These include contract version updates, naming specification adjustments, array length caching, function declaration optimizations, and data storage location adjustments. These modifications not only enhanced the code's conformity to standards and readability but also significantly improved the contract's execution efficiency and security. These improvements ensure the contract's robustness across different environments and align with industry best practices, laying a solid foundation for subsequent development and deployment.

Table 6. Results of contract analysis.

Contract	Error	Warn	Code Specification Advice
Administrator Contract	×	×	√
Student Contract	×	×	√
Teacher Contract	×	×	√
Course Contract	×	×	√
Work Contract	×	×	√
Regular Conditions Contract	×	×	√
Regular Scores Contract	×	×	√
Average of Regular Scores Contract	×	×	√
Midterm Scores Contract	×	×	√
Final Scores Contract	×	×	√
Comprehensive Scores Contract	×	×	√
Access Require Contract	×	×	√

In the table, the symbol '×' denotes that the issue is absent, whereas '√' denotes that it is present.

Table 7. Contract codifications.

Issues	Modifications	Whether Modified
Solc-0.5.0 version is outdated	Update to version 0.8.0 and revise the Solidity version in the smart contract	√
Contract naming does not follow the PascalCase (CapWords) specification	Capitalize the first letter of the contract name, ensuring no underscores or spaces are used	√
Variable naming does not adhere to the mixedCase (camelCase) specification	Ensure the first letter is lowercase, with each subsequent word's first letter capitalized	√
Loop condition should use cached array length	Cache the array length in a local variable within the function	√
Some functions should be declared "external"	Change the function declaration from "public" to "external"	√
Function parameters should change their data location	Change the storage location of function parameters from "memory" to "calldata"	√

In the table, the symbol '√' denotes that the issue has been modified.

8. Conclusions

This paper addresses the lack of timeliness and comprehensiveness in existing blended evaluation systems for recording learning achievements and rankings, as well as the limitations in the trusted storage and secure sharing of student work data. It studies and develops a blended teaching evaluation system based on blockchain and searchable encryption. The study first proposes a collaborative on-chain and off-chain storage model using the Ethereum blockchain and IPFS to enable large-scale, trusted storage of student work data. Next, it designs a role-based access control method based on smart contracts and introduces a searchable encryption scheme combining AES-CBC-256 and SHA-256 algorithms to ensure the secure sharing of student work. Additionally, the system records students' regular, midterm, final, and overall scores, along with their rankings, via smart contracts, ensuring transparency in the evaluation process. Finally, the system's gas consumption remains within the set limits, confirming its feasibility for practical application. In the future, we plan to promote the application of this system to provide practical solutions for teaching evaluation in various universities. On the other hand, we will improve the

functions, such as considering the implementation of fuzzy searchable encryption and attribute-based access control.

Author Contributions: Conceptualization, F.R.; methodology, F.R. and B.Z.; software, F.R. and T.-Y.X.; validation, F.R. and T.-Y.X.; formal analysis, B.Z. and J.-X.Z.; investigation, F.R.; resources, F.R. and T.Y-X.; data curation, T.-Y.X.; writing—original draft preparation, F.R. and T.-Y.X.; writing—review and editing, J.-X.Z. and J.W.; visualization, F.R. and T.-Y.X.; supervision, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by three grants: the first from the Project of the National Natural Science Foundation of China under Grant, grant number 62107034, and the second from the Major Science and Technology Project of Yunnan Province, grant number 202402AD080002. The third is from the Yunnan International Joint R&D Center of China-Laos-Thailand Educational Digitalization, grant number 202203AP140006.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ayasrah, F.T.M.; Shdouh, A.; Al-Said, K. Blockchain-based student assessment and evaluation: A secure and transparent approach in jordan's tertiary institutions. *Kurd. Stud.* **2023**, *11*, 2036–2049.
2. Karode, T.; Werapun, W. Performance analysis of trustworthy online review system using blockchain. In Proceedings of the 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, 24–27 June 2020; pp. 510–513.
3. Ouyang, S.; Huang, X. Education Evaluation Management Based on Blockchain Technology. *Mob. Inf. Syst.* **2022**, *2022*, 7513365. [CrossRef]
4. Ullah, Z.; Raza, B.; Shah, H.; Khan, S.; Waheed, A. Towards Blockchain-Based Secure Storage and Trusted Data Sharing Scheme for IoT Environment. *IEEE Access* **2022**, *10*, 36978–36994. [CrossRef]
5. Kumari, D.; Parmar, A.S.; Goyal, H.S.; Mishra, K.; Panda, S. Healthrec-chain: Patient-centric blockchain enabled ipfs for privacy preserving scalable health data. *Comput. Netw.* **2024**, *241*, 110223. [CrossRef]
6. Xie, T.; Yuan, L.; Zhang, Q.; Wu, J.; Ren, F. Ciphertext Fuzzy Retrieval Mechanism With Bidirectional Verification and Privacy Protection. *IEEE Internet Things J.* **2024**, *11*, 41061–41083. [CrossRef]
7. Song, D.X.; Wagner, D.; Perrig, A. Practical techniques for searches on encrypted data. In Proceedings of the Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000, Berkeley, CA, USA, 14–17 May 2000; pp. 44–55.
8. Chen, B.; Xiang, T.; He, D.; Li, H.; Choo, K.-K.R. BPVSE: Publicly Verifiable Searchable Encryption for Cloud-Assisted Electronic Health Records. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3171–3184. [CrossRef]
9. Senouci, M.R.; Benkhaddra, I.; Senouci, A.; Li, F. An efficient and secure certificateless searchable encryption scheme against keyword guessing attacks. *J. Syst. Archit.* **2021**, *119*, 102271. [CrossRef]
10. Fan, K.; Chen, Q.; Su, R.; Zhang, K.; Wang, H.; Li, H.; Yang, Y. MSIAP: A Dynamic Searchable Encryption for Privacy-Protection on Smart Grid With Cloud-Edge-End. *IEEE Trans. Cloud Comput.* **2023**, *11*, 1170–1181. [CrossRef]
11. Halverson, L.R.; Graham, C.R.; Spring, K.J.; Drysdale, J.S.; Henrie, C.R. A thematic analysis of the most highly cited scholarship in the first decade of blended learning research. *Internet High. Educ.* **2014**, *20*, 20–34. [CrossRef]
12. Kotkova, V.V. Blended learning course for future primary school teachers implementation. *Inf. Technol. Learn. Tools* **2017**, *60*, 312–323.
13. El-Ghalayini, H.; El-Khalili, N. An approach to designing and evaluating blended courses. *Educ. Inf. Technol.* **2012**, *17*, 417–430. [CrossRef]
14. Tesch, A. Implementing pre-post test designs in higher education evaluations. *New Dir. Eval.* **2016**, *151*, 85–96. [CrossRef]
15. Alammary, A. Blended learning models for introductory programming courses: A systematic review. *PLoS ONE* **2019**, *14*, e0221765. [CrossRef] [PubMed]
16. de Moura, V.F.; de Souza, C.A.; Viana, A.B.N. The use of Massive Open Online Courses (MOOCs) in blended learning courses and the functional value perceived by students. *Comput. Educ.* **2021**, *161*, 104077. [CrossRef]

17. Wu, X. Research on the reform of ideological and political teaching evaluation method of college English course based on "online and offline" teaching. *J. High. Educ. Res.* **2022**, *3*, 87–90. [CrossRef]
18. Li, X.; Yang, Y.; Chu, S.K.W.; Zainuddin, Z.; Zhang, Y. Applying blended synchronous teaching and learning for flexible learning in higher education: An action research study at a university in Hong Kong. *Asia Pac. J. Educ.* **2022**, *42*, 211–227. [CrossRef]
19. Widjaja, G.; Aslan, A. Blended learning method in the view of learning and teaching strategy in geography study programs in higher education. *Nazhruna J. Pendidik. Islam* **2022**, *5*, 22–36. [CrossRef]
20. Liu, X.Y.; Lu, C.; Zhu, H.; Wang, X.; Jia, S.; Zhang, Y.; Wen, H.; Wang, Y.F. Assessment of the effectiveness of BOPPPS-based hybrid teaching model in physiology education. *BMC Med. Educ.* **2022**, *22*, 217. [CrossRef]
21. Yang, Y.; Zhang, H.; Chai, H.; Xu, W. Design and application of intelligent teaching space for blended teaching. *Interact. Learn. Environ.* **2023**, *31*, 6147–6164. [CrossRef]
22. Gong, R.; Chen, Q. Research on application of cloud computing technology in blended learning based on access control model. In Proceedings of the 2021 2nd International Conference on Information Science and Education (ICISE-IE), Chongqing, China, 26–28 November 2021; pp. 471–474.
23. Cheriguene, A.; Kabache, T.; Kerrache, C.A.; Calafate, C.T.; Cano, J.C. NOTA: A novel online teaching and assessment scheme using Blockchain for emergency cases. *Educ. Inf. Technol.* **2022**, *27*, 115–132. [CrossRef]
24. Rahman, M.M.; Saha, U.K.; Islam, S.; Akhter, S. Blockchain-based Teaching Evaluation System for Ensuring Data Integrity and Anonymity. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 783–789. [CrossRef]
25. Chandratre, A.; Garg, S. Blockchain Based Course Feedback System. 2019. Available online: https://ssrn.com/abstract=3762332 (accessed on 3 March 2025).
26. Athanere, S.; Thakur, R. Blockchain based hierarchical semi-decentralized approach using IPFS for secure and efficient data sharing. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 1523–1534. [CrossRef]
27. Sultana, S.A.; Rupa, C.; Malleswari, R.P.; Gadekallu, T.R. IPFS-Blockchain Smart Contracts Based Conceptual Framework to Reduce Certificate Frauds in the Academic Field. *Information* **2023**, *14*, 446. [CrossRef]
28. Tong, W.; Yang, L.; Li, Z.; Jin, X.; Tan, L. Enhancing Security and Flexibility in the Industrial Internet of Things: Blockchain-Based Data Sharing and Privacy Protection. *Sensors* **2024**, *24*, 1035. [CrossRef]
29. Mahmoud, N.; Abdelkader, H.; Aly, A. BRIP: Towards a privacy-preserving, trustworthy, and transparent ride-sharing system with semantic matching powered by blockchain and IPFS. *J. Netw. Comput. Appl.* **2024**, *226*, 103870. [CrossRef]
30. Tao, X.; Das, M.; Liu, Y.; Cheng, J.C. Distributed common data environment using blockchain and Interplanetary File System for secure BIM-based collaborative design. *Autom. Constr.* **2021**, *130*, 103851. [CrossRef]
31. Andola, N.; Prakash, S.; Yadav, V.K.; Venkatesan, S.; Verma, S. A secure searchable encryption scheme for cloud using hash-based indexing. *J. Comput. Syst. Sci.* **2022**, *126*, 119–137. [CrossRef]
32. Li, H.; Huang, Q.; Huang, J.; Susilo, W. Public-Key Authenticated Encryption With Keyword Search Supporting Constant Trapdoor Generation and Fast Search. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 396–410. [CrossRef]
33. Pan, X.; Li, F. Public-key authenticated encryption with keyword search achieving both multi-ciphertext and multi-trapdoor indistinguishability. *J. Syst. Archit.* **2021**, *115*, 102075. [CrossRef]
34. Jiang, S.; Cao, J.; Wu, H.; Chen, K.; Liu, X. Privacy-preserving and efficient data sharing for blockchain-based intelligent transportation systems. *Inf. Sci.* **2023**, *635*, 72–85. [CrossRef]
35. Kushwaha, S.S.; Joshi, S.; Singh, D.; Kaur, M.; Lee, H.-N. Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract. *IEEE Access* **2022**, *10*, 6605–6621. [CrossRef]
36. Frantz, C.K.; Nowostawski, M. From institutions to code: Towards automated generation of smart contracts. In Proceedings of the 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W), Augsburg, Germany, 12–16 September 2016.
37. Kim, S.; Deka, G.C. *Blockchain Technology Use Cases*, 1st ed.; Kim, S., Deka, G.C., Eds.; Advanced Applications of Blockchain Technology; Springer: Singapore, 2020; Volume 60, pp. 91–114.
38. Laurent, A.; Brotcorne, L.; Fortz, B. Transaction fees optimization in the Ethereum blockchain. *Blockchain Res. Appl.* **2022**, *3*, 100074. [CrossRef]
39. Qiu, J.; Tian, Z.; Du, C.; Zuo, Q.; Su, S.; Fang, B. A Survey on Access Control in the Age of Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 4682–4696. [CrossRef]
40. Ferraiolo, D.F.; Sandhu, R.S.; Gavrila, S.I.; Kuhn, D.R.; Chandramouli, R. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.* **2001**, *4*, 224–274. [CrossRef]
41. Curtmola, R.; Garay, J.; Kamara, S.; Ostrovsky, R. Searchable symmetric encryption: Improved definitions and efficient constructions. In Proceedings of the 13th ACM conference on Computer and communications security, New York, NY, USA, 30 October–3 November 2006; pp. 79–88.
42. Pant, S.; Kumar, A.; Ram, M.; Klochkov, Y.; Sharma, H.K. Consistency indices in analytic hierarchy process: A review. *Mathematics* **2022**, *10*, 1206. [CrossRef]

43. Park, H.J. An exploration for the direction of emotion education in moral education based on the ultimate goal of moral education. *J. Moral Educ.* **2018**, *30*, 55–84. [CrossRef]
44. Cook, S.; Watson, D.; Webb, R. Performance evaluation in teaching: Dissecting student evaluations in higher education. *Stud. Educ. Eval.* **2024**, *81*, 101342. [CrossRef]
45. Ozkan, S.; Koseler, R. Multi-dimensional students' evaluation of e-learning systems in the higher education context: An empirical investigation. *Comput. Educ.* **2009**, *53*, 1285–1296. [CrossRef]
46. Babcock, P.; Marks, M. The falling time cost of college: Evidence from half a century of time use data. *Rev. Econ. Stat.* **2011**, *93*, 468–478. [CrossRef]
47. Finlay, M.J.; Simpson, T.; Tinnion, D.J. Association between attendance, online course activity time, and grades: Analysis of undergraduate sport science cohorts during the COVID-19 pandemic. *J. Hosp. Leis. Sport Tour. Educ.* **2022**, *31*, 100397. [CrossRef]
48. Gupta, P.; Mahajan, R.; Badhera, U.; Kushwaha, P.S. Integrating generative AI in management education: A mixed-methods study using social construction of technology theory. *Int. J. Manag. Educ.* **2024**, *22*, 101017. [CrossRef]
49. Liu, X. The educational resource management based on image data visualization and deep learning. *Heliyon* **2024**, *10*, e32972. [CrossRef] [PubMed]
50. Feist, J.; Grieco, G.; Groce, A. Slither: A Static Analysis Framework for Smart Contracts. In Proceedings of the 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Montreal, QC, Canada, 27 May 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.