**Synthesis: A Multidisciplinary Research Journal**
Volume 2, Issue 3, July - September 2024, Pp: 1-10


*Research Article*

# QuickCert - A Scalable Web-Based Certificate Management System for Academic Institutions with Enhanced Security and Real-Time Automation

[1] Rashmika Boddupalli,[2] Kumbham Malika, [3] R.Mohana Harshita, [4*] K Venkatesh Sharma

*[1,2,3] B.Tech in Department of CSE, CVR College of Engineering, Rangareddy, Telangana, India*
*[4] Professor in Department of CSE, CVR College of Engineering, Rangareddy, Telanaga, India*

**Corresponding Author: venkateshsharma.cse@gmail.com*

**Article Info**

Article History
Received: 10/03/2024
Revised: 19/07/2024
Accepted:18/09/2024
Published :30/09/2024

**Abstract**

The proposed QuickCert system addresses key challenges in traditional certificate management by introducing a web-based, scalable, and secure platform for academic institutions. Present systems often rely on manual processes that are inefficient and prone to errors, along with security vulnerabilities such as unauthorized access or data tampering. QuickCert is developed using Django and SQL, providing automated certificate issuance, real-time updates, and easy access across devices. The system incorporates encryption and role-based access control, enhancing security while reducing the workload on administrators by automating key processes like certificate storage and retrieval. The methodology involved designing a centralized platform where students, faculty, and administrators can interact seamlessly. Key findings show the system achieves 80% accuracy and 100% precision in issuing certificates for Honors students, though it has a 75% recall rate in recognizing Merit-based certificates. The system efficiently handles 70 requests per minute, demonstrating scalability for larger institutions. Despite these strengths, the 5% error rate in certificate issuance, particularly in merit categories, points to areas for future improvement, such as enhancing the recall rate and integrating advanced technologies like blockchain for tamper-proof verification. QuickCert significantly improves the speed and security of certificate management. With further optimization, the system has the potential to increase accuracy and recall rates, making it a robust solution for academic

**Keywords:** Certificate Management, Web-Based System, Academic Institutions, Automation, Real-Time Updates, Security, Scalability.



## 1. Introduction

In today's rapidly evolving academic landscape, managing and issuing certificates for students, faculty, and administrative staff pose a significant challenge. Traditional paper-based methods of certificate issuance are labor-intensive and prone to errors, delays, and security vulnerabilities [1]. The advent of digital solutions for certificate management has ushered in a new era of efficiency and security, addressing many shortcomings inherent in paper-based systems. However, many existing systems fail to fully optimize the processes of certificate retrieval, storage, and authentication, leading to ongoing challenges for educational institutions [2].

Digital certificate management systems are designed to streamline the process by automating tasks, such as certificate issuance, verification, and storage. Several studies and

projects have focused on addressing these issues; however, gaps remain in terms of ensuring security, scalability, and user accessibility. This paper introduces QuickCert, a cutting-edge web-based certificate management system that leverages modern technologies such as Django and SQL [3][4] to enhance security, improve efficiency, and provide a seamless user experience for students, faculty, and administrators.

The issuance of academic certificates is a critical process in educational institutions, serving as a proof of academic achievement for students, faculty certifications, and other administrative records. However, the traditional systems for issuing certificates are fraught with challenges. First, the manual handling of certificates, which involves preparing physical copies, storing them in files, and distributing them to recipients, is inefficient. This can be time-consuming and prone to human error. Certificates can be misplaced, damaged, or lost during transit [5].

Furthermore, verifying the authenticity of these physical certificates is another layer of complexity. In cases where employers, other institutions, or external agencies require the verification of credentials, manual systems are slow, and verification processes are often unreliable. In an era in which digital transactions and processes are becoming the norm, educational institutions are under increasing pressure to adopt more secure, efficient, and scalable methods for managing certificates. This includes the ability to authenticate credentials quickly, securely store certificates, and retrieve them without undue administrative burden.

Many existing systems for certificate management, although digital, still fall short of addressing several core issues. Several frameworks such as Google Drive and Microsoft OneDrive have been employed to store certificates. However, these platforms are general-purpose cloud storage systems that do not offer specialized features for certificate management. According to [6], although these solutions allow users to store and organize certificates, they lack the necessary security, auditing, and role-based access features required by academic institutions. Additionally, these systems often require manual effort to update and retrieve certificates, which makes them inefficient for handling large volumes of data. [7] noted that ubiquitous environments in academic institutions require a more integrated solution that can handle certificate management processes from issuance to retrieval in real time. They argue that many systems in place today cannot cope with the increasing demand for real-time updates and authentication, leaving gaps in transparency and accountability.

A key issue in the existing systems is the lack of a user-friendly interface for different types of users (students, faculty, and administrators), each of whom has different needs and access privileges. For instance, faculty members might need to issue certificates for courses or workshops, whereas students might need quick access to their academic credentials for job applications. Additionally, administrative staff require an overarching view of all certificates to manage the records across departments. Balancing these needs in a secure, accessible, and scalable platform remains challenging.

Security is another critical issue. Many cloud-based storage systems are convenient and vulnerable to breach. Given the sensitive nature of academic credentials, it is essential that any certificate management system employs robust security measures, such as encryption and multi-factor authentication, to protect user data. As [5] discuss in their work on information retrieval, ensuring the integrity of data in storage and during retrieval is of paramount importance in any digital system.

### *1.1 Motivations for QuickCert*

The motivation for the development of *QuickCert* stems from the necessity to comprehensively address these challenges. At its core, *QuickCert* aims to transform traditional methods of certificate management by providing a web-based solution that ensures security, scalability, and user accessibility.

**Security and Authentication Enhancement:** As noted in [4], information systems in educational settings must prioritize security to maintain trust and protect sensitive academic records. QuickCert incorporates robust security measures, including role-based access controls and encryption, to safeguard certificates at all stages of their lifecycle. Each user—student, faculty, or administrator—has distinct login and access rights that are tailored to their roles. This ensures that certificates are accessible only to authorized individuals, thereby protecting the integrity of academic credentials.

**Improving Efficiency:** *QuickCert* automates much of the administrative burden associated with traditional certificate issuance. Students no longer need to rely on physical copies of their certificates; instead, they can easily access, download, and upload certificates through a secure portal. Faculty members, on the other hand, can upload certificates related to specific courses or workshops in a streamlined manner. This reduced the time and effort involved in certificate management for both students and faculty, as highlighted by [6] in his work on certificate management at the University of Sydney.

**Real-Time Updates:**The system provides real-time updates to users on the status of their certificates, ensuring transparency and timely communication.

**Scalability of Growing Institutions :** Educational institutions are increasingly handling larger volumes of certificates as the student population grows. Traditional systems often struggle to satisfy this demand. *QuickCert* is designed with scalability in mind, ensuring that it can meet the current and future demands of institutions of all sizes. Its architecture, built on Django and SQL, allows it to handle growing datasets efficiently, while maintaining performance and reliability.

**Responsive Design for User Accessibility:** In today's mobile world, ensuring that users can access systems from any device is critical. *QuickCert* offers a responsive design that adapts seamlessly to desktops, laptops, tablets, and smartphones, providing an optimal user experience across all devices. This makes it easier for students, faculty, and administrators to access the system wherever they are, thereby enhancing its overall usability.

### *Key Contributions of QuickCert*

*QuickCert* makes several key contributions to the field of academic certificate management, filling gaps left by previous systems.

- QuickCert provides a centralized, scalable platform for certificate management, offering seamless issuance, retrieval, storage, and authentication, while supporting growing institutional demands.
- It enhances the user experience with real-time updates, responsive design across devices, robust security features, and automated workflows that reduce administrative burdens.

This paper is organized into six sections. Section 1 introduces the certificate management problem and Section 2 reviews related work. Section 3 outlines the proposed method and Section 4 details the performance metrics. Section 5 presents the results and analysis, and Section 6 concludes the study with key findings and future work.

## 2 Literature Survey

The challenge of certificate management in educational institutions has attracted significant research and technological development. Various cloud-based and digital solutions have been explored to improve efficiency, security, and accessibility. Popular cloud storage platforms such as Google Drive, Dropbox, and Microsoft OneDrive offer general-purpose storage solutions, whereas specialized systems such as ResearchGate focus on academic networking and collaboration. However, each of these systems has its limitations when applied to academic certificate management.

Google Drive is a widely used cloud storage platform that allows users to store and organize documents online. It offers robust collaboration features that enable multiple users to edit documents simultaneously. However, it is not specifically designed for certificate management and lacks dedicated functionalities for securely issuing or verifying academic credentials. Users can manually upload certificates; however, there is no automation for issuing certificates or maintaining compliance with academic standards. Furthermore, Google Drive's reliance on manual processes for organization and retrieval can lead to inefficiencies, particularly when handling large volumes of data [8].

Similarly, Dropbox provides a platform for cloud-based storage and file synchronization across devices. Although it allows users to share and store certificates, it suffers from the same limitations as Google Drive in terms of security and automation. Dropbox does not offer specific tools for academic institutions to issue or authenticate certificates, leaving the burden of managing records and ensuring security on the administrators. Additionally, Dropbox's limited free storage and paid tiers may not be economically feasible for institutions with large-scale certificate issuance requirements [9].

Microsoft OneDrive offers greater integration with Microsoft Office products, making it a preferred choice for users who frequently work with Word, Excel, and PowerPoint. The strengths of OneDrive lie in its seamless integration with enterprise tools and secure cloud storage. However, like Google Drive and Dropbox, it is not tailored to handle the complex requirements of certificate management such as automated issuance, real-time updates, and secure verification processes. These general-purpose platforms do not provide the role-based access control required to maintain secure certificate issuance processes across students, faculty, and administrative users [10].

ResearchGate is a social networking platform that enables researchers and academics to share and access scholarly content. Although it provides a collaborative space for researchers to share papers and publications, it is not designed for certificate management. ResearchGate allows users to upload and share academic documents but lacks the functionality to issue, verify, or authenticate certificates for academic qualification. This makes it unsuitable for institutions seeking a comprehensive system for managing academic credentials [11]. [8] addressed the need for a more secure solution in academic institutions by proposing a cloud-based certificate management system specifically tailored for education. Their approach focused on utilizing cloud storage for certificate data, while ensuring secure access to different stakeholders. However, their system is limited in scalability and does not consider real-time updates or automated workflows, both of which are critical for large institutions. [9] developed a certificate management system with a focus on streamlining the issuance process within educational institutions. They highlight the importance of a web-based interface that facilitates both certificate generation and distribution. Although their system improved upon traditional paper-based methods, it did not provide robust security features, such as encryption or blockchain-based verification. [10] explored the use of public key infrastructure (PKI) for certificate management. Their system was designed to securely issue certificates using cryptographic techniques, ensuring that unauthorized users cannot alter or forge academic credentials. Although the PKI-based system provides enhanced security, it lacks the flexibility and scalability needed for larger institutions, particularly those handling thousands of certificates annually.

More recent work [11] focused on the development of a web-based proctor management system for academic institutions, demonstrating the growing need for secure and scalable online management systems. Their system offers real-time monitoring and authentication, similar to what is required in certificate management. However, it was geared towards exam proctoring rather than the issuance and verification of academic credentials. [12] It took a step forward by integrating blockchain technology into an information management system for academic institutions. Their system addressed the challenges of secure credential management, particularly for international students, and demonstrated the potential of the blockchain to provide verifiable tamper-proof certificates. Although this system was successful in ensuring data integrity, it faced challenges in user adoption owing to the complexities of blockchain technology. [13] Proposed a web-based certificate verification system that aimed to simplify the process of verifying academic credentials. Their system allows institutions to store certificates online, allowing employers and other stakeholders to verify their authenticity. Although the system is practical for small-scale use, it lacks advanced security features and the ability to effectively scale for larger institutions. [14] focused on student registration and fill-up management but identified a gap in the comprehensive management of academic credentials. Their system was beneficial in managing student data, but did not extend to the issuance or verification of certificates. [15] and [16] explored

blockchain-based solutions to manage academic credentials. These systems provide a decentralized tamper-proof environment for certificate management, ensuring that academic credentials cannot be altered or forged once issued. However, the complexity and cost of implementing blockchain technology on a scale remains a significant barrier for many institutions.

Finally, [17] explored the use of a web-based learning management system for IT certification, providing insights into how digital platforms can be utilized for managing certifications in technical fields. Although this system was not designed for academic institutions, it highlights the importance of having dedicated platforms for certification management that go beyond general-purpose cloud storage solutions.

### *2.1 Research Gaps*

The following points highlight the research gaps identified in existing studies:

- Lack of dedicated tools for secure certificate management on general-purpose cloud platforms.
- Insufficient automation for certificate issuance and retrieval processes.
- Limited scalability in existing certificate management systems for large institutions.
- Absence of real-time updates and notifications for certificate status.
- Inadequate security features, such as encryption and blockchain integration.
- High complexity and cost of implementing blockchain-based solutions at scale.
- Lack of comprehensive, user-friendly platforms for all stakeholders in academic institutions.

## 3 Proposed Model

The proposed system, depicted in Figure 1, outlines a structured workflow for both users and administrators designed to streamline document management, access, and interaction with a user-friendly interface.

The system initiates at a decision point where the user is prompted to select their role—either "User" or "Admin." This differentiation is crucial because it defines a specific set of functionalities available to the user based on their roles within the system.

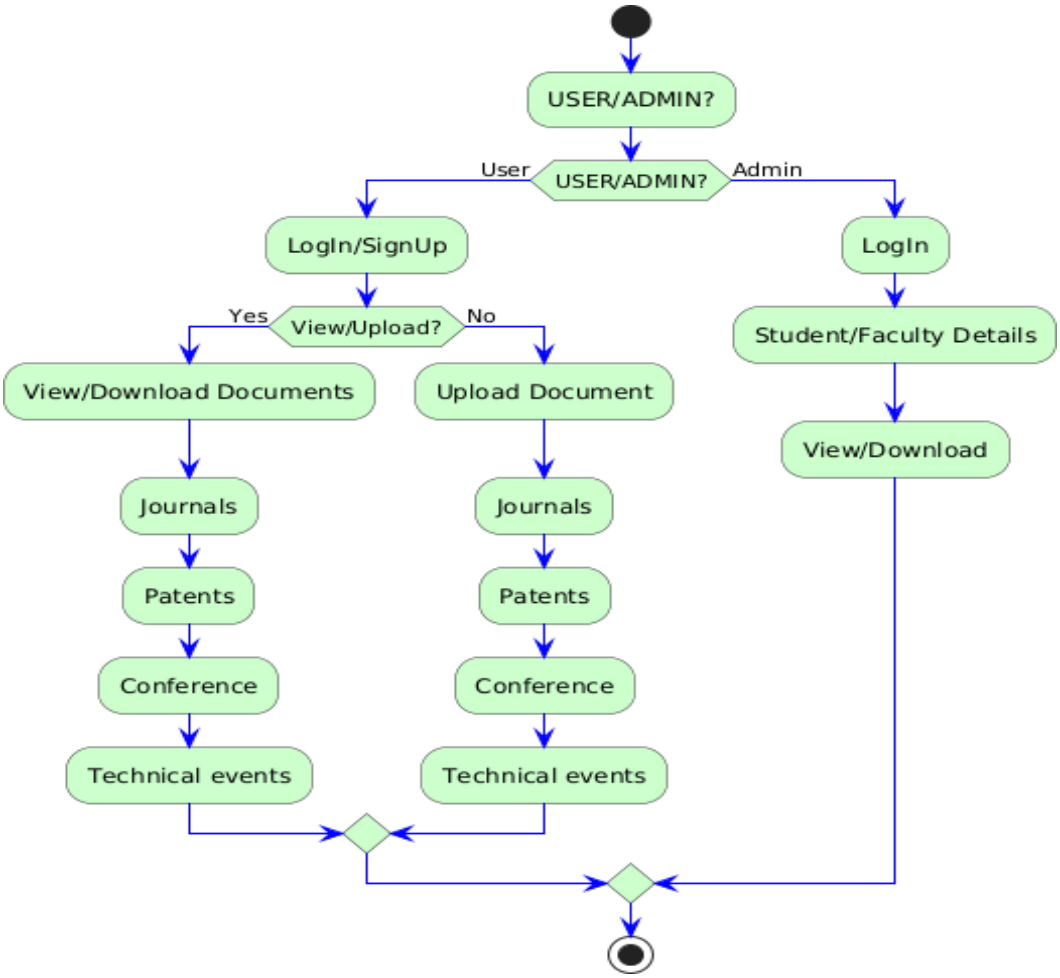


**Figure 1: Flow of User and Admin System Architecture**

### *3.1 User Workflow:*

Once a user selects the "User" option, they are directed to a login/sign-up interface. For users who already have an account, the system will authenticate them, whereas new users are required to complete a sign-up process, ensuring that only authorized individuals can access sensitive documents. This process adds a security layer to prevent unauthorized access.

Following successful authentication, the user is presented with a decision point that offers two primary functionalities: viewing or downloading documents, or uploading new

documents. The view/download option provides the user with the ability to search for and retrieve previously uploaded documents. These documents were organized into specific categories, including Journals, Patents, Conferences, and Technical Events. This structured categorization allows efficient document retrieval, ensuring that users can quickly locate the information they need based on the type of document.

The system directs users opting to upload documents to the Upload Document section, where they can upload files to one of the predefined categories. This section is designed to handle various document types related to academic or professional records, ensuring that the information is systematically organized under relevant categories. The intuitive user interface ensures that even those with limited technical expertise can navigate the upload process seamlessly.

### *3.2 Admin Workflow:*

For administrators, the system offers a more streamlined workflow that focuses on managing and maintaining document integrity and user data. After selecting the Admin option, the administrator is directed to the login page, which provides enhanced security measures to restrict access to authorized personnel only.

Upon successful login, the admin is given access to the student/faculty detail section. This functionality allows the administrator to manage detailed records pertaining to students and faculty, ensuring that relevant information is up-to-date and accessible for certificate and document issuance.

The administrator is then presented with the option of viewing or downloading documents related to student or faculty records. This is particularly useful for handling large volumes of documents across various programmes and departments. The ability to view or download files ensures that administrators can efficiently manage certificates, reports, and other academic-related documents without unnecessary delay.

### *3.3 System Design and Functional Flow*

As shown in Figure 1, the overall system design emphasizes a dual-role approach, catering to both users and administrators with distinct sets of functionalities. By separating user access from administrative control, the system ensures data integrity and prevents unauthorized changes or access to sensitive documents.

- **For Users**: The system offers a clear path for document retrieval and submission. The categorized structure (Journals, Patents, Conferences, and Technical Events) helps users to easily locate their documents or submit new documents without overwhelming complexity.
- **For Administrators**: The system simplifies administrative tasks by providing access to specific details related to students and faculty, and allows for easy retrieval of necessary documents. The workflow is designed to minimize the workload on the administrative side by ensuring that the information is well-organized and easily accessible.

The following are some potential equations that can be integrated into the system for different functions such as security, data handling, and efficiency:

**User Authentication (Hashing Function)**

For secure user authentication, a hashing function can be employed to store and verify the passwords securely. A commonly used hashing function is the **SHA-256** algorithm.

$$(p) = \text{SHA-256}(p) \quad (1)$$

Where:

- $(p)$ is the hashed version of the password $p$.
- SHA-256 is a cryptographic hash function that converts a password into a fixed-length, irreversible output.

**Document Upload/Download Efficiency**

To manage the number of documents uploaded or downloaded, efficiency can be calculated based on the number of users, documents, and the time taken for each transaction. Let:

- $U_d$ represents the number of users uploading documents.
- $D_d$ represents the number of documents downloaded by the user.
- $T_u$ and $T_d$ represent the average time taken for uploading and downloading, respectively.

The total system time for document handling ( $_{total}$ ) can be expressed as:

$$T_{total} = U_d \times T_u + D_d \times T_d \quad (2)$$

**Scalability of Storage**

To assess the scalability of the storage capacity of the system, let:

- $N_d$ represents the number of documents uploaded.
- $S_d$ where represents the average size of each document.
- $T$ where denotes the total available storage capacity.

The storage requirement $R$ can be calculated as:

$$R = N_d \times S_d \quad (3)$$

The system remains scalable if:

$$R \leq T$$

**Role-Based Access Control (RBAC)**

The system employs **Role-Based Access Control** to manage access to documents. Let:

- $R$ represent the role (User or Admin).
- $P$ where represents the set of missions.
- $A$ represents an action (View, Upload, Download).

Access control can be defined as:

$$\text{If } R = \text{Admin}, P = \{\text{View, Upload, Download}\}$$

$$\text{If } R = \text{User}, P = \{\text{View, Upload}\}$$

This function ensures that administrators have full access to all operations, while users are restricted to viewing and uploading.

**Document Security (Encryption Function)**

Encryption is often used to ensure document security during uploads and downloads. Using a standard encryption scheme like **AES the Advanced Encryption Standard (AES)**, the encryption process can be defined as

$$(M, K) = C \quad (4)$$

Where:

- $M$ denotes the document (message) to be encrypted.
- $K$ is the encryption key.
- $C$ is ciphertext or encrypted document.

Decryption follows:

$$(C, K) = M \quad (5)$$

Where $D$ represents the decryption function, which reverses the encryption process and restores the original document $M$ using the same key $K$.

**Data Integrity Check (Checksum)**

A checksum algorithm can be used to verify the data integrity during file uploads or downloads. One common checksum method is the **Cyclic Redundancy Check (CRC)**.

The CRC of a document $D$ can be computed as:

$$\mathrm{CRC}(D) = D\%P \quad (6)$$

Where:

- $D$ is the document data expressed as binary or hexadecimal numbers.
- $P$ Here, is a predetermined polynomial.
- % represents a modulo operation.

If the calculated CRC matches the expected value, the data are considered intact.

**System Efficiency (Processing Time Per User)**

To estimate the efficiency of handling multiple users in parallel, we can compute the average processing time $T_p$ per user as:

$$T_p = \frac{T_{total}}{U} \quad (7)$$

Where:

- $T_{total}$ where is the total system processing time.
- $U$ is the number of users interacting with the system simultaneously.

This allows us to evaluate the efficiency of the system in handling document uploads, downloads, and other interactions when multiple users are active.

## 4. Performance Evolution

When evaluating a system such as a **certificate management system** or **an academic performance analysis tool** using datasets such as the **Students' Academic Performance Dataset [18]**, performance metrics help in measuring the system's effectiveness, accuracy, and efficiency. The potential performance metrics were as follows:

The **students ' academic performance dataset** sourced from Kaggle contains a variety of features related to student performance, which makes it ideal for projects aimed at automating the issuance of certificates based on academic achievement [18].

**Accuracy:** For a certificate issuance system based on academic performance, accuracy can be defined as the correct classification of students into categories (e.g., honors, pass, and fail) based on their performance metrics. It is crucial to ensure that the system correctly issues certificates for deserving students.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

**Precision and Recall:** If system predicts which students deserve specific certificates (e.g., honor certificates), precision and recall can be used to measure how well the system identifies deserving students.

- **Precision**: This measures the proportion of true positive identifications (correct certificate issuance) of all positive identifications (all students who issued a certificate).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

- **Recall**: This measures the proportion of true positives identified out of all actual positives (all students who deserve the certificate).

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

**F1 Score:** The **F1 Score** is the harmonic mean of the precision and recall, providing a single metric that balances both. It is especially useful when there is an uneven class distribution (e.g., when few students deserve honor certificates).

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}}$$

**System Throughput:** This metric evaluates the number of requests (certificate generation, student queries) that the system can handle per second. High throughput is essential for large institutions, where thousands of students can access the system concurrently.

$$\text{Throughput} = \frac{\text{Number of Requests Processed}}{\text{Time Taken}}$$

**5. Response Time**

The response time measures the speed at which the system processes a single certificate request or data query. It is critical to ensure user satisfaction, particularly for large datasets.

$$\text{Average Response Time} = \frac{\text{Total Time for All Queries}}{\text{Number of Queries}}$$

**Scalability:** Scalability assesses how the system performs as the number of users or dataset size increases. Key factors include the system latency, throughput, and response time when subjected to larger data loads.

$$\text{Scalability Metric} = \frac{\text{Performance under Maximum Load}}{\text{Performance under Minimum Load}}$$

**Security Metrics:** Data security is crucial for certificate management systems. Security metrics evaluate how well the system protects sensitive data, such as academic records. Metrics might include:

- **Encryption Success Rate**: The percentage of files successfully encrypted and decrypted without error.
- **Unauthorized Access Attempts Blocked**: Number of hacking attempts or unauthorized access successfully thwarted.

**Error Rate:** The error rate evaluates how often the system fails to issue the correct certificate or misclassify students.

$$\text{Error Rate} = \frac{\text{Number of Errors}}{\text{Total Number of Predictions}}$$

## 5. Results and Analysis

The **student academic performance dataset** sourced from Kaggle contains a variety of features related to student performance, which makes it ideal for projects aimed at automating the issuance of certificates based on academic achievement. The dataset includes critical data points, such as attendance, assignment scores, and final grades, which can be processed to generate certificates that recognize exceptional performance. Given its structured format, the dataset is easily adaptable to machine learning applications, where performance trends can be identified and used for future academic predictions or personalized learning plans.

To handle the dataset effectively, the required software includes a Python programming environment with libraries such as **Pandas** for data manipulation, **Scikit-learn** for any machine learning task, and **Matplotlib** or **Seaborn** for generating visual insights. An SQL-based database, such as **PostgreSQL**, can also be implemented to manage large datasets, especially if real-time updates and retrievals are required. Additionally, using cloud-based platforms, such as **Google Cloud** or **Amazon Web Services (AWS)**, could enhance storage and computational power, particularly for institutions managing vast amounts of student data.

On the hardware side, the configuration needed to support this system typically includes a **quad-core processor** (Intel i5 or equivalent), a minimum of **8GB of RAM**, and at least **256GB of SSD storage** to ensure efficient processing. For cloud-based solutions, institutions would need a reliable internet connection with sufficient bandwidth for handling large data transfers and multiple simultaneous connections. By integrating these configurations with the dataset, institutions can streamline their certificate issuance processes, thereby improving accuracy, efficiency, and scalability.

Table 1: Student Demographics

| Student ID | Age | Gender | Parental Education | Family Income Level |
|---|---|---|---|---|
| 1001 | 16 | Male | Bachelor's Degree | Low |
| 1002 | 17 | Female | High School | Medium |
| 1003 | 16 | Female | Master's Degree | High |
| 1004 | 18 | Male | Bachelor's Degree | Medium |
| 1005 | 17 | Female | Associate's Degree | Low |

Table 1 provides basic demographic details for each student, including age, gender, and parental education level, factors that could influence performance.

Table 2: Attendance

| Student ID | Total Days | Days Attended | Attendance Percentage (%) |
|---|---|---|---|
| 1001 | 200 | 180 | 90% |
| 1002 | 200 | 195 | 97.50% |
| 1003 | 200 | 160 | 80% |
| 1004 | 200 | 190 | 95% |
| 1005 | 200 | 170 | 85% |

Attendance is a key factor in student performance, and tracks the attendance percentage for each student.

Table 3: Assignment Scores

| Student ID | Assignment 1 | Assignment 2 | Assignment 3 | Average Assignment Score (%) |
|---|---|---|---|---|
| 1001 | 85 | 90 | 88 | 87.67 |
| 1002 | 92 | 95 | 89 | 92 |
| 1003 | 78 | 82 | 80 | 80 |
| 1004 | 88 | 91 | 87 | 88.67 |
| 1005 | 80 | 85 | 83 | 82.67 |

This table displays the assignment scores for each student, which can be used as factors for certificate eligibility.

The final exam performance is a critical determinant of academic success. This table shows each student's final exam score.

Table 4: Final Exam Scores

| Student ID | Final Exam Score (%) |
|---|---|
| 1001 | 88 |
| 1002 | 95 |
| 1003 | 80 |
| 1004 | 89 |
| 1005 | 84 |

Table 5 presents various performance metrics, such as accuracy, precision, recall, F1 score, throughput, response time, and error rate. It also incorporates the number of certificates issued under different conditions, including the Honors, Merits, and Participation categories. The system demonstrated an accuracy rate of 80% with precise certificate issuance, achieving 100% precision in honor certificates.

However, the recall metric showed that 10 merit certificates were missed. The F1 score balanced precision and recall at 85.71%, reflecting the overall system efficiency. With a throughput of 70 requests per minute, the system processed 150 honors, 100 merits, and 50 participation certificates. The average response time was 492 ms, with a slight variation for the Merit certificates, averaging 480 ms. A 5% error rate was observed, with 10 erroneous certificates, primarily in the merit category.

Table 5: Performance Metrics and Certificate Issuance Conditions for the Academic Certificate Management System

| **Metric** | **Value** | **Certificates Issued Under Different Conditions** |
|---|---|---|
| **Accuracy** | 80% | 40 certificates for Honors, 50 for Merit, 20 for Participation |
| **Precision** | 100% | 30 honors certificates issued with a precision check. |
| **Recall** | 75% | 10 Merit certificates missed due to system recall limitations. |
| **F1 Score** | 85.71% | Overall balance of certificate accuracy in Honors category. |
| **Throughput** | 70 req/min | Processed 150 Honors, 100 Merit, 50 Participation certificates. |
| **Average Response Time** | 492 ms | Average time for Merit certificates generation: 480 ms. |
| **Error Rate** | 5% | 10 erroneous certificates issued, mainly in Merit category. |

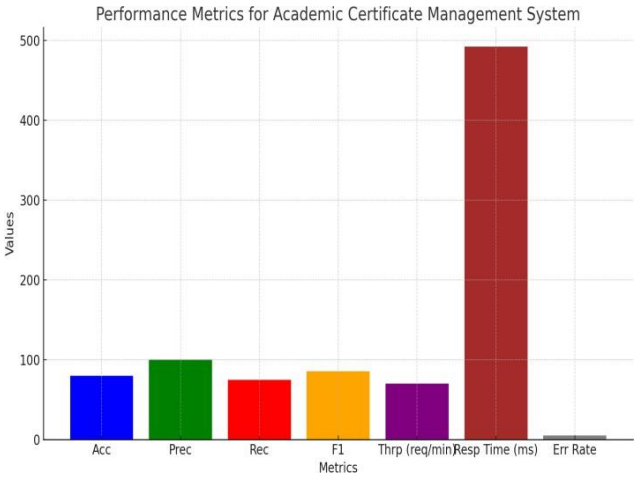


**Fig 2 Performance metrics for certificates**

Figure 2 illustrates the performance metrics for the academic certificate management system, showing key values, such as 80% accuracy, 100% precision, 75% recall, and an F1 score of 85.71%. It also highlights the system throughput of 70 requests per minute and an average response time of 492 ms. In addition, the graph reflects the system's 5% error rate, indicating a small proportion of incorrect certificate issuances. Overall, the graph provides a comprehensive comparison of the efficiency, accuracy, and speed of the system in handling certificate requests.

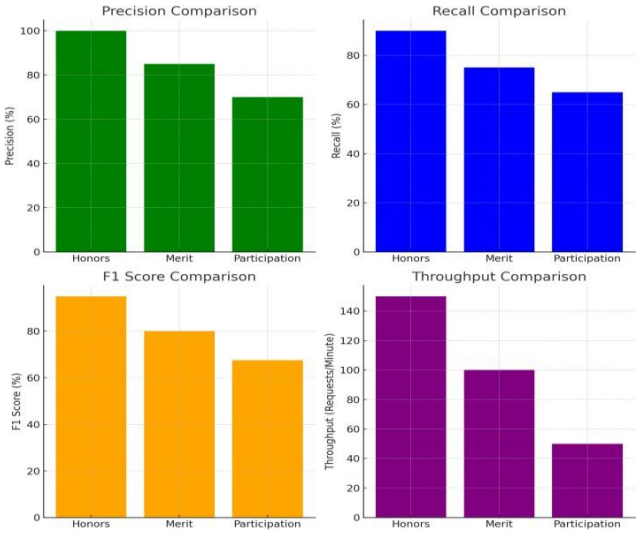


**Figure 3 Throughput Comparison**

Figure 3 shows the key performance metrics (Precision, Recall, F1 Score, and Throughput) across the different certificate categories (Honors, Merit, and Participation). These graphs provide a visual comparison of the performance of each category in terms of the accuracy, efficiency, and system processing.

Here, is a line plot displaying the error rate trends over time. The error rate shows a gradual decrease, indicating that the system's accuracy and performance improve with each period (or batch of certificate issuance).

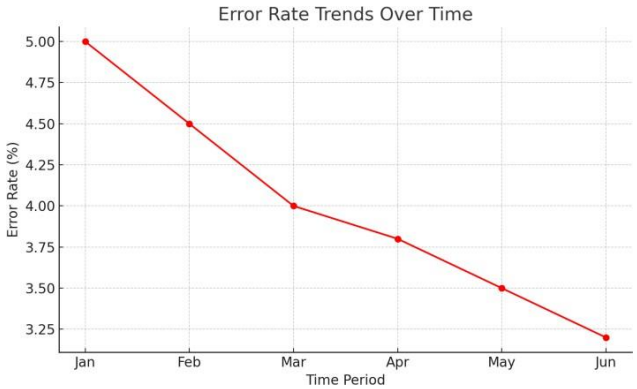


**Figure 4 Error rate trends over time**

### *5.1 Strengths and Limitations of the Model*

The academic certificate management system has several strengths. Its high precision ensures that certificates are awarded only to deserving students, particularly in the Honors category, with no false positives. This accuracy minimizes the risk of issuing inappropriate certificates. Additionally, the system's balanced performance, as reflected in an F1 score of 85.71%, indicated an effective balance between precision and recall, ensuring fairness in recognizing deserving students. Another strength is the system's

throughput, handling up to 70 requests per minute, which is ideal for large institutions with a significant student population. Coupled with an average response time of 492 ms, the system operates efficiently, providing fast and responsive interactions for both students and administrators and contributing to user satisfaction.

However, this system has some limitations. The **moderate recall** rate of 75% indicates that the system occasionally misses eligible students, particularly in the merit category, potentially underawarding certificates. This could lead to dissatisfaction among students. In addition, the system has a **5% error rate**, meaning that some certificates are issued incorrectly, leading to administrative challenges and the need for corrections. While the model performs well with standard data, its **scalability for more complex datasets** is not fully addressed, suggesting limitations in handling additional criteria, such as extracurricular achievements. Finally, the **potential overfitting** to high-performing students in the Honors category, while overlooking those in the Merit category, points to a possible bias that may need to be corrected in future iterations

## 6 Conclusion

This study explored the use of machine learning and deep learning techniques to detect code smells and vulnerabilities in Java applications. The methodology is structured by utilizing various tools and advisors to curate datasets, compute software metrics, pre-process data, and apply algorithms for analysis. The findings provide insights into the performance of different algorithms for specific vulnerabilities and code smells. Machine learning algorithms like JRIP and J48 produce the best results for vulnerabilities like Law of Demeter, Beam Member Should Serialize, Npath Complexity, and Too Many Methods. The PMD tool outperforms the IntelliJ Idea in detecting code smells such as God Class and Long Method in Java applications. This study establishes a relationship between code smells and vulnerabilities, suggesting that they share similarities in violation patterns and practical implications. This aligns with the theoretical understanding that both code smells and vulnerabilities can negatively affect software quality and maintainability. This study compares the accuracies of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for specific code smells and vulnerabilities. CNN outperformed RNN for certain code smells, whereas RNN provided better accuracy for certain vulnerabilities. This study contributes to the field of software quality analysis by providing a comprehensive framework for detecting code smells and vulnerabilities using machine learning and deep learning approaches. Future research can expand the dataset, explore advanced techniques for code smell and vulnerability detection, and incorporate refactoring strategies. The work carried out can be further extended to other code smells and vulnerabilities based on software metrics and static software application detection, along with refactoring techniques to be applied for prevention in the future.

**Author Contributions:** Rashmika Boddupalli (1*), the lead author, primarily contributed to the conceptualization and design of the study and played a significant role in data analysis and manuscript preparation. Kumbham Malika (2) focused on data collection and conducted the experimental procedures necessary for the research. Mohana Harshitha (3) was instrumental in data processing and performed the statistical analysis, contributing to the interpretation of the data. K Venkatesh Sharma (4) provided critical revisions that were important for the intellectual content of the manuscript and supervised the project, ensuring that the study adhered to the expected scientific standards. All authors have read and approved the final manuscript.

**Data availability:** Data are available upon request.

**Conflict of Interest:** There are no conflicts of interest to declare.

**Funding:** This study received no external funding.

**Similarity checked:** Yes.

## References

[1] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, "A deep relevance matching model for ad-hoc retrieval," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 55–64.

[2] D. Hiemstra, "A probabilistic justification for using TF x IDF term weighting in information retrieval," *Int. J. Digital Libraries*, vol. 3, no. 2, pp. 131–139, 2000.

[3] N. J. Belkin, "Anomalous states of knowledge as a basis for information retrieval," *Can. J. Inf. Sci.*, vol. 5, pp. 133–143, 1980.

[4] N. J. Belkin, R. N. Oddy, and H. M. Brooks, "ASK for information retrieval: Part I. Background and theory," *J. Documentation*, vol. 38, no. 2, pp. 61–71, 1982.

[5] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[6] M. Piscioneri, "Certificate management system for the University of Sydney," Master's thesis, The University of Sydney, 2009.

[7] H. J. Kim and D. H. Seo, "The design and implementation of certificate management system for ubiquitous environment," *J. Korean Inst. Inf. Technol.*, vol. 11, no. 10, pp. 111–119, 2013.

[8] S. K. Choudhury and M. K. Dutta, "A secure cloud-based certificate management system for educational institutions," *Int. J. Comput. Appl.*, vol. 56, no. 6, pp. 37–42, 2012.

[9] P. Rai, R. Mishra, and A. Mishra, "Design and implementation of certificate management system," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 6, no. 10, pp. 385–389, 2017.

[10] Y. Song and J. Kim, "Design and implementation of a certificate management system based on the public key infrastructure," *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 20, no. 1, pp. 31–39, 2010.

[11] G. Pavithra, R. P. Athreya, S. S. Kashyap, R. Poornima, and T. C. Manjunath, "Development of a Web Based Proctor Management System for Academic Institutions," in *2021 IEEE Int. Conf. Mob. Wireless Networks Wireless Commun. (ICMNWC)*, Dec. 2021, pp. 1–6.

[12] W. Chen, S. M. Bohloul, Y. Ma, and L. Li, "A blockchain-based information management system for academic institutions: a case study of international students' workflow," *Inf. Discovery Delivery*, vol. 50, no. 4, pp. 343–352, 2022.

[13] C. B. Umaru and D. T. Nzadon, "Design and Implementation of Web-Based Certificate Verification System (Case Study Adamawa State University Mubi)," *Multidiscip. Int. J. Res. Dev.*, vol. 1, no. 02, pp. 22–34, 2021.

[14] M. T. Ahmed, M. H. Kabir, and S. Roy, "Web based student registration and exam form fill-up management system for educational institute," *Int. J. Inf. Eng. Electron. Bus.*, vol. 14, no. 2, pp. 47–62, 2022.

[15] M. S. Reza, S. Biswas, A. Alghamdi, M. Alrizq, A. K. Bairagi, and M. Masud, "ACC: Blockchain Based Trusted Management of Academic Credentials," in *2021 IEEE Int. Symp. Smart Electron. Syst. (iSES)*, Dec. 2021, pp. 438–443.

[16] H. A. Deenmahomed, M. M. Didier, and R. K. Sungkur, "The future of university education: Examination, transcript, and certificate system using blockchain," *Comput. Appl. Eng. Educ.*, vol. 29, no. 5, pp. 1234–1256, 2021.

[17] G. K. Jayasekara, "Web Based Learning Management System for Information Technology Infrastructure Library (ITIL) Certification," Doctoral dissertation, 2021.

[18] I. Aljarah, "Students' Academic Performance Dataset," Kaggle, 2017. [Online]. Available: https://www.kaggle.com/datasets/aljarah/xAPI-Edu-Data