Submission

📋 My Files

🖥 My Files

🎓 University

Document Details

Submission ID

trn:oid:::28592:80603570

Submission Date

Jan 31, 2025, 12:53 AM GMT+5:30

Download Date

Jan 31, 2025, 12:54 AM GMT+5:30

File Name

746533-Script.docx

File Size

23.0 KB

6 Pages

1,213 Words

7,884 Characters

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

1 AI-generated only 0%
Likely AI-generated text from a large-language model.

2 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

Slide 1: Introduction

- Manual grading creates inefficiencies
- Paper/spreadsheet methods cause errors (Basir et al., 2024).
- C++ system automates grade tracking
- Secure, scalable, and efficient
- Supports role-based access control

Speaker Notes:

A proper and safe grading system is required in educational institutions. Due to invalidating the conventional manual grading system, one of its outcomes is delays, errors and liability concerns (Basir et al., 2024). The purpose of the C++ Student Grade Management System is to automate the grading processes. It is integrated into the database and has role-based access for facilitating quick and secure data handling (Fareed & Yassin, 2022). It offers a structured, error-free grading process and automatic analysis of the student's performance. It helps increase the accuracy of the data and decision-making within educational institutions.

Slide 2: Problem Statement

- Manual grading causes inefficiencies
- High error rate in calculations
- Lack of security for student data (Fareed & Yassin, 2022).
- No real-time performance analytics
- Needs secure digital transformation

Speaker Notes:

Paper records and spreadsheets are also the means that the majority of traditional grading systems rely upon, therefore error-prone and inefficient (Basir et al., 2024). However, many educators suffer from inaccuracy and timeliness, which often results in miscalculations. This necessitates the absence of strong security measures that allow student data to be vulnerable to unauthorized access (Fareed & Yassin, 2022). Moreover, teachers do not have real-time analytic insights into students' performance trends. In addressing these issues of grading student works, this project digitizes accuracy, security and efficiency while doing so.

Slide 3: System Design Decisions

- Uses object-oriented programming
- C++ ensures performance efficiency
- SQLite database for secure storage
- Authentication with encrypted passwords
- Data analysis for performance insights

Speaker Notes:

C++ is an efficient object-oriented programming language that the system is built using. This is to make it easy to be modular, scalable, and reusable (Ali, 2024). The primary database used for storing and fast retrieval of student records is SQLite (Scientific, 2024). In light of this, the system would integrate the mechanisms of authentication with the hashing with the encrypted password in order to safeguard the confidentiality of data(Kokila & Reddy, 2024). It has built-in analytics tools to give an education of performing trends, which help in data-driven decision making. Thus, it will be a robust, future-proof system.

Slide 4: Implementation Details

- CRUD operations manage student data
- Secure role-based access control
- Fast retrieval with optimized queries (Khataei, 2024).
- ASCII graphs visualize performance
- Effective error handling integrated

Speaker Notes:

The system allows one to perform create, read, update, and delete operations that make student data management very efficient. Access control based on the role is also restricted in order to restrict unwanted access and protect sensitive student records (Fareed & Yassin, 2022). This helps the system optimize SQLite queries and shortens processing time, increasing performance (Khataei, 2024). In this case, ASCII graphs assist educators in analyzing student performance trends more easily. Furthermore, the system comprises advanced error-handling mechanisms to make the system stable and reliable while sending feedback to the users.

Slide 5: Infrastructure Requirements

- Works on Windows, Mac, Linux
- Requires C++ and SQLite setup (Birkenkrahe, 2023).
- Uses Code::Blocks, VS Code IDEs
- Database Browser for management
- Supports multi-platform accessibility

Speaker Notes:

The system is a cross-platform on Windows, Mac, and Linux. It relies on C++ compilers and SQLite database performance (Birkenkrahe, 2023). The main development environment has

been set as Code::Blocks and Visual Studio Code to make structured programming and effective debugging. Database management is done for sorted structured and secure storage using SQLite Database Browser. The system is designed for up scalability and multi-platform usability and, hence, can be used by different educational institutions.

Slide 6: Evaluation & Testing

- Unit tests for data accuracy
- Speed tests for database queries
- Security tests for authentication
- System stability through debugging
- User experience enhancement tested

Speaker Notes:

The system's accuracy, performance, and security are verified by comprehensive testing. Student records are validated using unit tests to ensure that calculation errors do not occur (Maswanganyi et al., 2024). The database speed tests verify quick query execution so that there is an efficient application performance. Authentication measures are tested for security testing, where only the users who are authorized should be able to access the student data (Fareed & Yassin, 2022). Debugging the system leads to an increase in stability and user experience. These tests confirmed that the system operates properly in real educational settings.

Slide 7: Future Work

- Develop graphical user interface (Paneru et al., 2024).
- Implement cloud-based data storage
- Integrate AI for student analytics
- Automate personalized learning insights
- Expand scalability for institutions

Speaker Notes:

Although the system is working to meet user needs, a better user experience and capabilities are planned for the near future. The system will be enhanced by a graphical user interface (GUI) instead of the command line interface (Paneru et al., 2024). This will improve cloud integration, allowing remote access and automatic synchronization of student records. Student performance trends will be analyzed, and the result will be personalized recommendations for struggling students based on artificial intelligence. Furthermore, the scalability will permit the easy use of the system by institutions with a large number of students.

Slide 8: Conclusion

- C++ ensures performance efficiency
- Database enhances secure record storage
- Secure authentication prevents breaches
- AI will enhance analytics further
- System scalability supports growth

Speaker Notes:

This project shows how C++ and SQLite are efficient, secure, scalable systems for student grade management. The system optimizes the grading operations by reducing the errors and improving the security. Another security role-based authentication makes sure that student's records are safe. The system will remain an important tool for educators and administrators in academic institutions due to the continued refinement of the system. Finally, improvements in the future, like AI-powered analytics and cloud-based solutions, would help the system to perform better.

References

Ali, Q. I. (2024). Towards more effective summative assessment in OBE: a new framework integrating direct measurements and technology. *Discover Education*, *3*(1), 107. https://doi.org/10.1007/s44217-024-00208-5

Basir, M. S., Buckmaster, D., Raturi, A., & Zhang, Y. (2024). From pen and paper to digital precision: a comprehensive review of on-farm recordkeeping. Precision Agriculture, 25(5), 2643-2682. https://doi.org/10.1007/s11119-024-10172-7

Birkenkrahe, M. (2023). Teaching Data Science with Literate Programming Tools. Digital, 3(3), 232-250. https://doi.org/10.3390/digital3030015

Fareed, M., & Yassin, A. A. (2022). Privacy-preserving multi-factor authentication and role-based access control scheme for the E-healthcare system. *Bulletin of Electrical Engineering and Informatics*, *11*(4), 2131-2141. https://doi.org/10.11591/eei.v11i4.3658

Khataei, S. (2024). A BIM-Integrated Agent-Based Simulation Method for Time-Space Conflict Detection Among Mobile Resources in Construction Projects (Doctoral dissertation, Middle East Technical University (Turkey)). https://www.proquest.com/openview/803528dfc5930fc37c660bc395d26859/1?pq-origsite=gscholar&cbl=2026366&diss=y

Kokila, M., & Reddy, S. (2024). Authentication, Access Control and Scalability models in Internet of Things Security-A Review. Cyber Security and Applications, 100057. https://doi.org/10.1016/j.csa.2024.100057

Maswanganyi, N., Fumani, N., Khoza, J. K., Thango, B., & Lerato, M. (2024). Evaluating the impact of database and data warehouse technologies on organizational performance: A systematic review. *Available at SSRN 4997368*. https://dx.doi.org/10.2139/ssrn.4997368

Paneru, B., Paneru, B., Poudyal, R., & Shah, K. B. (2024). Exploring the Nexus of User Interface (UI) and User Experience (UX) in the Context of Emerging Trends and Customer Experience, Human Computer Interaction, Applications of Artificial Intelligence. International Journal of Informatics, Information System and Computer Engineering (INJIISCOM), 5(1), 102-113. https://doi.org/10.34010/injiiscom.v5i1.12488

Scientific, L. L. (2024). Data Integration Approaches And Data Classification Algorithms: A Review. *Journal of Theoretical and Applied Information Technology*, *102*(17). https://www.jatit.org/volumes/Vol102No17/16Vol102No17.pdf