Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (Print) | 2583-7907 (Online)
Journal Homepage: http://ajcst.co
DOI: https://doi.org/10.70112/ajcst-2026.15.1.4373

Research Article

Enhancing Fairness and Efficiency in Subjective Assessment through LLM-Based Automated Grading

A. Sanusi Funmilayo[1*], Lucas-Adebayo Daniel[2], Fatade Oluwayemisi Boye[3] and Okorie Grace Chinenye[4]
[1,4]Department of Software Engineering, [3]Department of Computer Science, Babcock University, Ilisan-Remo, Ogun State, Nigeria
[2]Department of Computer Science and Mathematics, Mountain Top University, Ogun State, Nigeria

Article Information

Article History
Received: 17 September 2025
Revised: 5 October 2025
Accepted: 27 November 2025
Published online: 5 January 2026

Keywords
Automated Grading Systems
Large Language Models
Vision-Based Document Analysis
Educational Assessment
Multi-Agent Architectures

Correspondence*
sanusifu@babcock.edu.ng

ORCID
A. Sanusi Funmilayo
https://orcid.org/0000-0001-5794-3657
Lucas-Adebayo Daniel
https://orcid.org/0009-0005-3496-2480
Fatade Oluwayemisi Boye
https://orcid.org/0000-0002-8514-9802
Okorie Grace Chinenye
https://orcid.org/0009-0006-0216-6911

Abstract

In recent years, the demand for fairness, speed, and transparency in grading has catalyzed interest in automated systems, particularly for subjective, theory-based assessments. Unlike objective tests, these examinations require nuanced understanding and contextual reasoning, traditionally making them dependent on human graders. However, human grading is often affected by inconsistencies, biases, and fatigue-induced errors. This work presents a system that leverages Large Language Models (LLMs) as grading agents for automating the evaluation of handwritten, theory-based exam scripts. Methods: The methodology employs a modular system architecture in which uploaded scripts are digitized, interpreted using vision-based models, and subsequently graded by domain-specific LLM agents. The system is implemented using FastAPI for the backend, Celery and RabbitMQ for asynchronous task handling, Redis for log streaming and task status management, and Next.js for the frontend interface. For mathematics scripts, a Math Agent is used to evaluate exam responses through context-aware reasoning. Results: Preliminary evaluation indicates that the system can grade an eight-question script within three minutes, significantly faster than the approximately fifteen minutes required by a human grader. This demonstrates that LLM-based grading systems can scale efficiently while reducing human bias and fatigue. Discussion and Conclusion: The project provides a foundation for broader integration of LLMs into educational assessment, while acknowledging limitations in current open-source vision models and inference latency. Future improvements may include fine-tuning and offline model support to enhance speed and reliability.

I. INTRODUCTION

The landscape of education is continuously evolving, with increasing student populations and the proliferation of online learning platforms posing significant challenges to traditional assessment methods [1]. Manual grading of theory-based exams, while valuable for providing nuanced feedback, is often time-consuming and resource-intensive, leading to delays in student feedback and increased workloads for educators [2], [3]. Furthermore, traditional grading practices are susceptible to subjectivity and inconsistencies, potentially affecting the fairness and reliability of assessment outcomes, as referenced in [3]. In response to these challenges, automated assessment systems have emerged as a promising solution. Early automated grading techniques relied on Natural Language Processing (NLP) methods to analyze student responses. These methods, including rule-based systems and statistical approaches, demonstrated some success in grading objective question types. However, they often struggled to accurately evaluate the complex reasoning, conceptual understanding, and critical thinking required in theory-based exams—particularly those involving subjective assessments [2]. The advent of Large Language Models (LLMs) has opened new avenues for automating the evaluation of subjective responses. LLMs possess advanced capabilities in understanding and generating human language, offering the potential to overcome many limitations associated with traditional NLP-based grading

systems [4]. Although LLMs have shown promising results across various assessment tasks [3], challenges remain, including potential biases, limitations in explainability, and the need for domain-specific adaptation. This paper seeks to contribute to advancements in automated assessment by developing a system that leverages LLMs for grading theory exams. Recognizing the diversity of academic disciplines and their unique requirements, this project adopts a multi-agent approach in which individual agents are designed to address subject-specific nuances. The initial focus is on developing an agent for mathematics, with the long-term objective of expanding the system to additional subjects. The system is intended to support teachers and educational institutions by improving the efficiency, accuracy, and scalability of theory exam grading in both traditional and online learning environments.

A. Problem Statement

Current methods for grading theory exams present significant challenges. Manual grading is time-consuming and resource-intensive, resulting in increased educator workloads and delays in providing feedback to students. Subjectivity in grading can lead to inconsistencies and concerns regarding fairness. Traditional automated grading systems have difficulty accurately evaluating the complex reasoning and conceptual understanding required in subjective assessments. While LLMs offer notable improvements, they also bring challenges such as potential biases, limited explainability, and the need for domain-specific adaptation. These issues hinder the scalability, efficiency, and reliability of automated theory-exam grading.

B. Objective and Scope

The primary aim of this paper is to develop and evaluate an automated multi-agent grading system that leverages LLMs to assess theory-based examination responses across multiple subjects. The specific objectives are to:

1. Define the functional and non-functional requirements of an automated grading system for theory-based exams, focusing on the use of LLMs for subject-specific assessment.
2. Design a scalable multi-agent system architecture in which each subject is managed by a dedicated grading agent equipped with LLM-based logic.
3. Develop and implement a proof-of-concept grading agent for mathematics theory exams using LLMs and prompt-engineering techniques.
4. Conduct a performance evaluation of the mathematics grading agent by comparing its results with human grading benchmarks in terms of accuracy and grading time.
5. Extend the architecture by implementing agents for additional subjects, demonstrating system adaptability and modularity.
6. Analyze the effectiveness, scalability, and fairness of the proposed multi-agent system in automating theory exam grading at scale.

The scope of this study encompasses the design and development of an automated multi-agent system tailored for grading theory-based examination scripts. Central to this project is the creation of a modular architecture in which individual agents are responsible for subject-specific grading tasks. For demonstration, the implementation focuses on a single subject—mathematics—while the underlying architecture is developed to accommodate additional agents in future iterations. The grading process leverages LLMs in conjunction with prompt-engineering techniques. System performance is assessed by comparing automated results with human examiner evaluations. The system is intended as a practical tool for educators and institutions in both traditional and online learning environments. However, the project does not include grading of objective-type questions such as multiple-choice or fill-in-the-blank formats. Additionally, developing a full production-level user interface is outside the scope, as the emphasis is on validating core automated grading functionality.

C. Literature Review

There is increasing global emphasis on fairness, speed, and transparency in grading, leading to growing interest in automating grading processes—particularly for subjective and theory-based assessments. These examinations require nuanced understanding, reasoning, and interpretation, making them difficult to grade consistently and objectively. While human grading provides contextual and domain expertise, it is often affected by inconsistency, fatigue-induced errors, and unconscious biases. Automated systems, by contrast, offer scalability, reliability, and standardization. Recent advancements in artificial intelligence (AI), especially LLMs, have opened new opportunities for automating the evaluation of subjective academic content. This section reviews the evolution of automated grading technologies, from rule-based and statistical approaches to modern deep-learning-based models.

D. Traditional Automated Grading Techniques

Early developments in automated grading focused primarily on objective formats such as multiple-choice or cloze tests. These formats were compatible with simple computational strategies that did not require deep language comprehension. Rule-based systems, the earliest attempts to simulate human scoring logic, relied on pre-programmed criteria such as keyword presence or grammatical structure. Reference [1] describes these systems as effective in constrained environments but inherently rigid, often penalizing valid alternative answers. Statistical approaches, which examined features such as word frequency, sentence length, and grammatical accuracy, followed. These metrics were processed using regression models or decision trees to predict scores. As demonstrated in [5], such techniques could

approximate human scoring under specific conditions but failed to evaluate semantic richness or conceptual depth. Students could use complex vocabulary without conveying the correct meaning, resulting in misleading scores. Moreover, these systems lacked adaptability across subjects or question formats, highlighting the need for more flexible, semantically aware models.

E. Natural Language Processing (NLP) in Educational Assessment

Integrating NLP into educational technology marked a significant advancement over earlier rule-based methods. NLP enabled systems to analyze linguistic structures through tokenization, part-of-speech tagging, dependency parsing, and semantic role labeling. These methods allowed systems to detect plagiarism, provide automated suggestions, and analyze sentiment [1].
Despite these improvements, NLP remained largely syntactic rather than semantic. It could identify *what* was written but struggled to evaluate *why* it was written or how effectively it addressed the question. Thus, while NLP provided a foundation, it did not fully solve the challenges associated with grading subjective assessments.

F. Large Language Models: Foundations and Advancements

The introduction of LLMs marked a paradigm shift in language processing. Based on the Transformer architecture introduced in [6], LLMs leverage attention mechanisms to capture long-range dependencies across text. Unlike earlier task-specific models, LLMs are pre-trained on massive corpora, enabling them to learn complex linguistic patterns, reasoning capabilities, and instruction-following behavior.
Models such as GPT-3, GPT-4, Claude, and PaLM have demonstrated the ability to answer complex questions, summarize documents, and generate human-like essays. Their capacity for zero-shot and few-shot learning allows them to perform tasks without extensive retraining. In educational contexts, LLMs can evaluate student writing, detect logical errors, and simulate peer review. As explained in [3], their versatility positions them as powerful tools for subjective assessment.

G. LLMs for Subjective Exam Scoring: Potential and Limitations

Recent studies indicate that LLMs can match or even surpass human graders under certain conditions. References [7] and [8] found that GPT-4 achieved human-level agreement in 70–93% of essay-scoring scenarios. LLMs excel at identifying key concepts, evaluating coherence, and recognizing well-structured arguments. However, limitations persist. LLMs often demonstrate conservative scoring tendencies, avoiding extremely high or low marks. As indicated in [3], this score centrality arises from probabilistic averaging behavior learned during pre-training. Explainability also remains a concern, with many frameworks lacking transparent justification for assigned scores. Domain-specific variation further complicates deployment, as LLMs perform inconsistently across disciplines requiring interpretive reasoning. Additionally, reliance on rigid rubric-based templates can hinder adaptation to unexpected or nuanced responses.

H. Challenges and Limitations of LLM-Based Automated Grading

Key challenges include:
1. Bias and Fairness: Inherited biases from training data may influence scoring [3].
2. Higher-Order Thinking: Current models struggle to evaluate originality and abstract reasoning [4].
3. Manipulability: Students may exploit keyword-based or stylistic patterns to influence scores [4].
4. Ethics and Privacy: Use of student data raises concerns regarding consent and protection [1].
5. Transparency: Lack of clear explanations can make scores difficult to justify or contest.

I. Subject-Specific Considerations

Different academic disciplines present unique grading challenges. Mathematics theory questions often require evaluation of logical sequences, symbolic notation, and procedural correctness. Understanding derivations and equivalent formulations is essential [2]. Conversely, literature or history responses demand analysis, argumentation, and evidence synthesis, requiring sensitivity to tone, factual accuracy, and interpretive nuance [9]. Tailoring AI grading agents to subject-specific requirements is a promising approach [10]. For example, a math agent may integrate symbolic parsers, while a humanities agent may incorporate rhetorical analysis modules. Developing such modular, explainable agents is a core motivation behind this project.

II. RELATED WORKS

The growing demand for automation in education has driven significant research into smart grading systems. These systems leverage artificial intelligence (AI), natural language processing (NLP), and machine learning (ML) to improve efficiency, consistency, and feedback in student assessment. One major stream of research focuses on automated essay scoring (AES). Darwish *et al.* [11] proposed a neutrosophic ontology-based engine to score English essays, addressing uncertainties in semantic interpretation. Similarly, Li *et al.* [12] introduced a multi-scale feature approach that integrates both local and global features for more accurate grading. A systematic review by Darwish *et al.* [13] synthesized multiple AES methods, revealing the predominance of NLP and statistical models while highlighting gaps in fairness and contextual adaptability. Recent work has also examined the use of large language models (LLMs) such as GPT-4, PaLM 2, and Claude 2 for automated essay scoring. Although these models demonstrate strong validity and reliability, concerns remain about consistency and bias, particularly in evaluating

English-language learners [14]. Beyond essay writing, researchers have explored feature-rich approaches. Alawadh *et al.* [15] developed a text-mining framework incorporating lexical statistics and discourse macro-structures, achieving high predictive accuracy. In related work, Sun *et al.* [16] combined neural networks with Random Forest classifiers not only to grade English writing tasks but also to provide automated feedback. In technical education, research has expanded into STEM and programming assessments. The GRAD-AI system by Gambo *et al.* [17] offers automated grading for programming assignments by executing student code against test cases and generating targeted feedback. A complementary review by Messer *et al.* [18] classified automated programming graders, discussing their strengths, limitations, and integration challenges. Ahmed and Sorour [19] further proposed an intelligent system for exam paper evaluation, measuring quality in alignment with Bloom's taxonomy and linking grading with pedagogical objectives.

Recent studies have also emphasized dataset development. Akinwale and Tunde-Adeleke [20] introduced a structured dataset specifically designed for automated grading research, addressing the scarcity of domain-specific evaluation data, particularly within African higher education contexts. Although not directly focused on grading, work in educational automation provides valuable insights. For example, Eweoya *et al.* [21] developed a university attendance management system using geofencing. Their design principles—automation, authentication, and system reliability—offer transferable lessons for building robust and trustworthy grading systems. Overall, the literature demonstrates strong progress in smart grading; however, persistent gaps remain. These include limited focus on multimodal assessment (integrating code, essays, and diagrams), a lack of culturally and linguistically localized datasets, insufficient attention to fairness and explainability, and limited integration with existing learning management systems (LMS). Addressing these gaps is essential for the development of reliable and context-sensitive smart grading systems.

III. MATERIALS AND METHODS

The methodology adopted for this research is fundamentally constructive in nature, aligning with software engineering projects aimed at producing novel, functional systems. This approach is well suited for real-world application domains where new software artifacts must be developed and empirically validated.
The key phases of the constructive methodology include:

1. *Problem Identification:* Manual grading of theory examinations is time-consuming, error-prone, and inconsistent across evaluators.
2. *Artifact Design and Construction:* Development of a multi-agent, LLM-driven grading engine that emulates human evaluation strategies.
3. *Implementation:* Iterative prototyping involving prompt engineering, LangGraph flow design, and real-time system feedback mechanisms.

A. System Architecture and Data Flow

The system employs a modular, multi-agent architecture in which independent LLM-based agents handle various stages of the grading workflow, as depicted in Figure 1. This architecture is designed to be scalable, maintainable, and capable of supporting multiple subjects simultaneously. The workflow begins with scanned examination scripts, which serve as the raw input to the system. These scanned documents are processed by a Multimodal Extraction Agent, which utilizes a vision-language model capable of jointly interpreting text and layout. Unlike traditional approaches that rely on separate Optical Character Recognition (OCR), segmentation, and parsing modules, this unified method simplifies processing and improves accuracy.

The agent extracts structured data, including:
1. Question text
2. Student answers

The extracted data is then stored in a centralized, queryable database for downstream processing.

Next, relevant question–answer pairs are passed to the Subject Grading Agents (beginning with the Mathematics Agent). These agents evaluate responses using prompt-engineering techniques. Prompts are guided by uploaded rubrics, which assist the LLM in score allocation and justification generation. LangGraph serves as the coordination backbone, ensuring stateful management of agent outputs and handling task routing between workflow nodes.

Each subject agent is configured with domain-specific logic. For the Mathematics Agent:
1. The prompt includes the grading rubric, sample model answers, and scoring criteria.
2. The input includes extracted student responses, question prompts, and allocated marks per question.
3. The output includes the assigned score and a brief justification.

B. Multi-Agent Integration via LangGraph

LangGraph orchestrates communication and task transitions between all components in the system. It models the grading pipeline as a directed graph, where each node represents a processing task or an intelligent agent. Nodes are connected by conditional transitions based on system states.

For example:
1. *Start node:* Script input and initialization
2. *Extraction node:* Multimodal document analysis
3. *Math node:* Executes the Mathematics Agent
4. *Fallback node:* Triggers re-evaluation or manual review upon failure

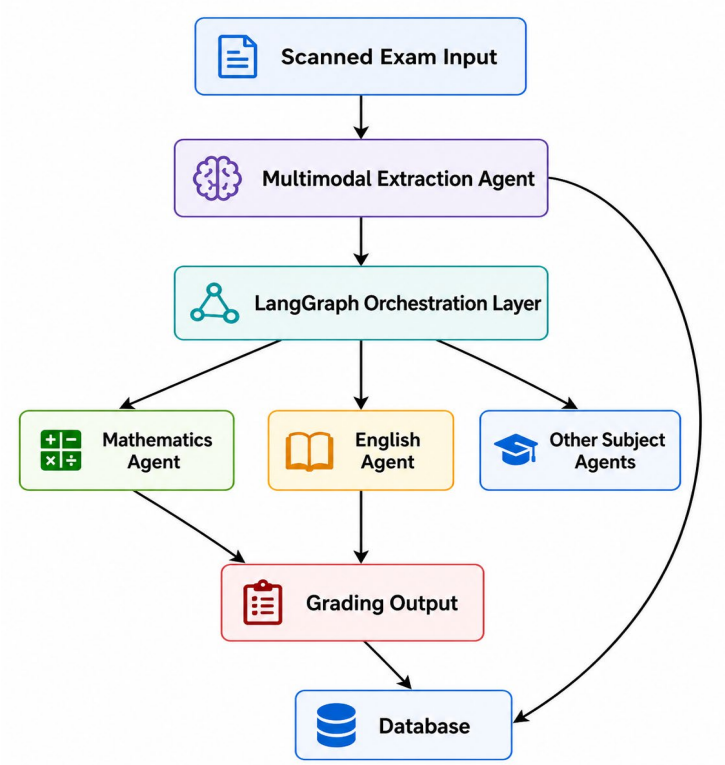

Fig.1 System Architecture Overview Showing Interaction Between Agents and Orchestration Layer

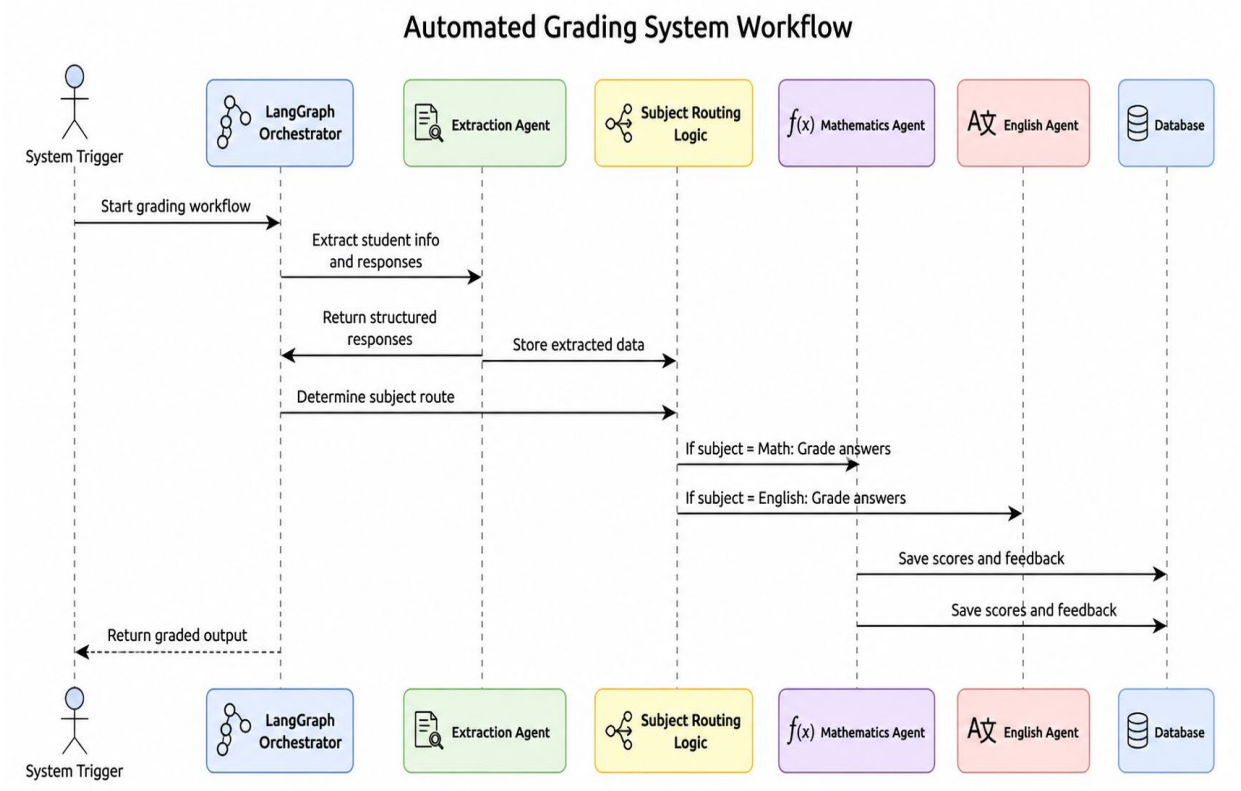

Fig.2 Grading Process Flowchart Detailing Routing Logic Based on Subject Type

A key strength of LangGraph is its dynamic adaptability. When a new subject agent (e.g., English or Biology) is introduced, it is added as an additional node and the transition rules are updated accordingly. This allows seamless scaling without disrupting the overall architecture. Furthermore, LangGraph supports parallel execution. If a script contains

responses from multiple subjects, the corresponding agents can grade their sections concurrently, thereby improving efficiency. Error handling is integrated into the design.

If an agent times out, returns malformed output, or expresses uncertainty, LangGraph can reroute the task, re-prompt the LLM, or log the case for human intervention. LangGraph transforms the grading system from a collection of isolated AI tasks into a cohesive, intelligent workflow. It provides structured control, fail-safety, and modular scalability, enabling seamless coordination of complex grading processes across multiple subjects with minimal manual intervention. Figure 2 and Figure 3 show the grading process flowchart and the UML sequence diagram of the multi-agent grading workflow managed by LangGraph.

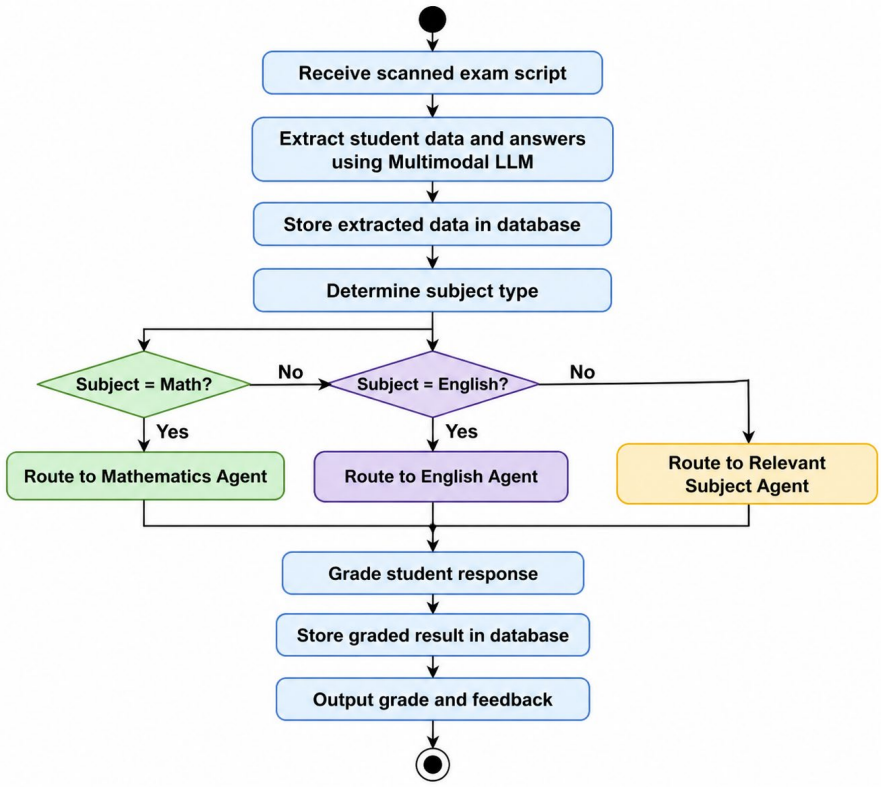

Fig.3 UML Sequence Diagram of Multi-Agent Grading Workflow Managed by LangGraph

The system is designed as a full-stack web application with the following major components:

1. *Frontend:* Developed using Next.js with TypeScript and styled using Tailwind CSS and Shadcn/UI components.
2. *Backend:* Built using FastAPI (Python) and LangGraph. FastAPI handles HTTP and WebSocket communication, whereas LangGraph manages the flow of grading agents. Together, they orchestrate grading logic, task dispatching, and result delivery.
3. *Asynchronous Processing Layer:* Powered by Celery for background task execution, with RabbitMQ as the message broker and Redis as the result backend.
4. *Data Stores:*
   a. SQLite for persistent data (via Prisma ORM)
   b. Redis for caching and real-time task tracking

Users create grading sessions through the frontend, which are then transmitted to the backend for processing. The system asynchronously grades each paper and streams real-time updates and results back to the frontend. Table I presents the system's functional requirements, Table II outlines the non-functional requirements, and Table III lists the technologies and tools used in designing the system.

TABLE I FUNCTIONAL REQUIREMENTS (FR) OF THE GRADING SYSTEM

Requirement ID	Description
FR1	System shall accept scanned theory exam scripts as input.
FR2	System should extract questions, student responses, and metadata (name, subject) using a vision-LLM.
FR3	System shall route extracted data to the appropriate subject-specific grading agent.
FR4	Each grading agent shall evaluate responses using predefined rubrics and prompt templates.
FR5	System shall use LangGraph to orchestrate the workflow (extraction → grading → storage).
FR6	The system should generate and store grading outcomes in a structured format (e.g., database, JSON).

TABLE II NON-FUNCTIONAL (NFR) REQUIREMENTS OF THE GRADING SYSTEM

Requirement ID	Description
NFR1	Grading accuracy shall be within ±10% of human-assigned scores based on rubric alignment.
NFR2	System should support adding new subject agents without major architectural changes (scalability).
NFR3	Agents shall be independent modules, allowing individual upgrades or replacements (modularity).
NFR4	Student data and results must be securely stored and accessed only by authorized users (security/privacy).

TABLE III TECHNOLOGIES AND TOOLS USED IN THE GRADING SYSTEM

Component	Technology	Justification
Frontend	Next.js, TypeScript, Tailwind, Shadcn/UI	Fast, reactive UI with reusable components
Backend	FastAPI, Python, LangGraph	High-performance async APIs and LLM orchestration
Task Queue	Celery	Handles concurrent grading jobs
Message Broker	RabbitMQ	Reliable task queue messaging
Cache/Task Store	Redis	Real-time tracking of tasks
Database	SQLite + Prisma ORM	Lightweight yet sufficient for prototyping
LLM SDK	OpenAI SDK	Seamless API access to hosted models

IV. RESULTS

Figures 3, 4, 5, and 6 show the grading session creation page, the grading process view, the result dashboard, and the session logs obtained via WebSocket communication, respectively. Figure 7 shows the interface of the results and feedback model.

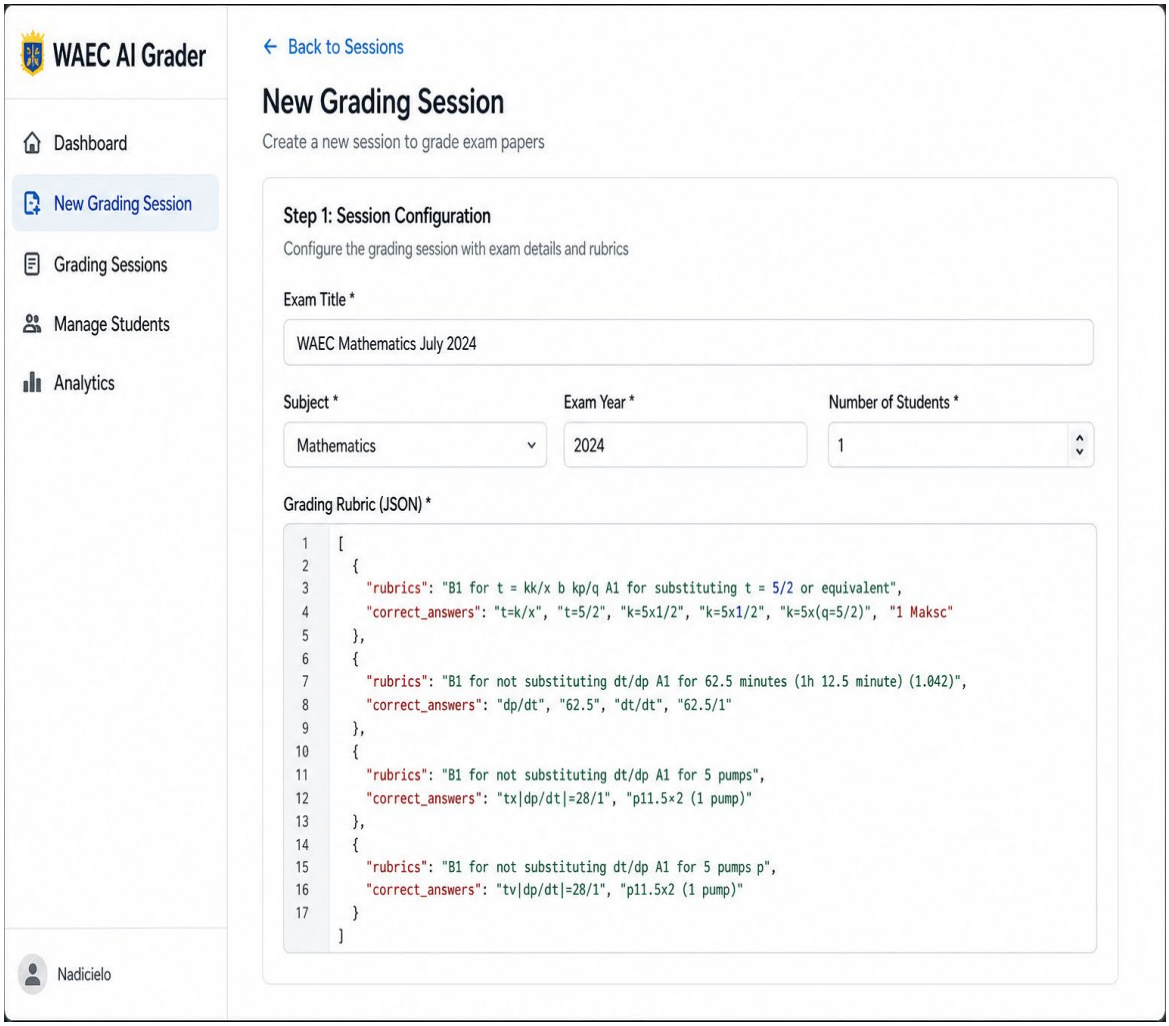

Fig.4 Image Showing the Grading Session Configuration Page

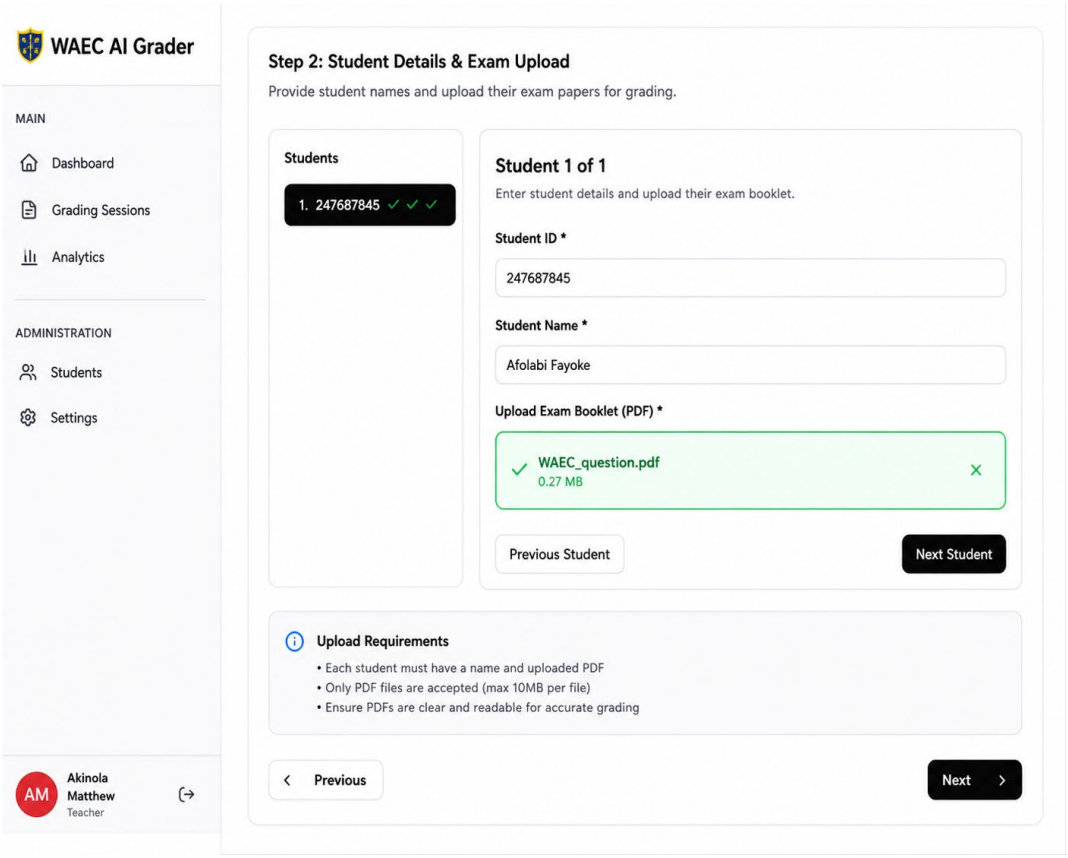

Fig.5 Image Showing the Student Details and Exam Upload Page

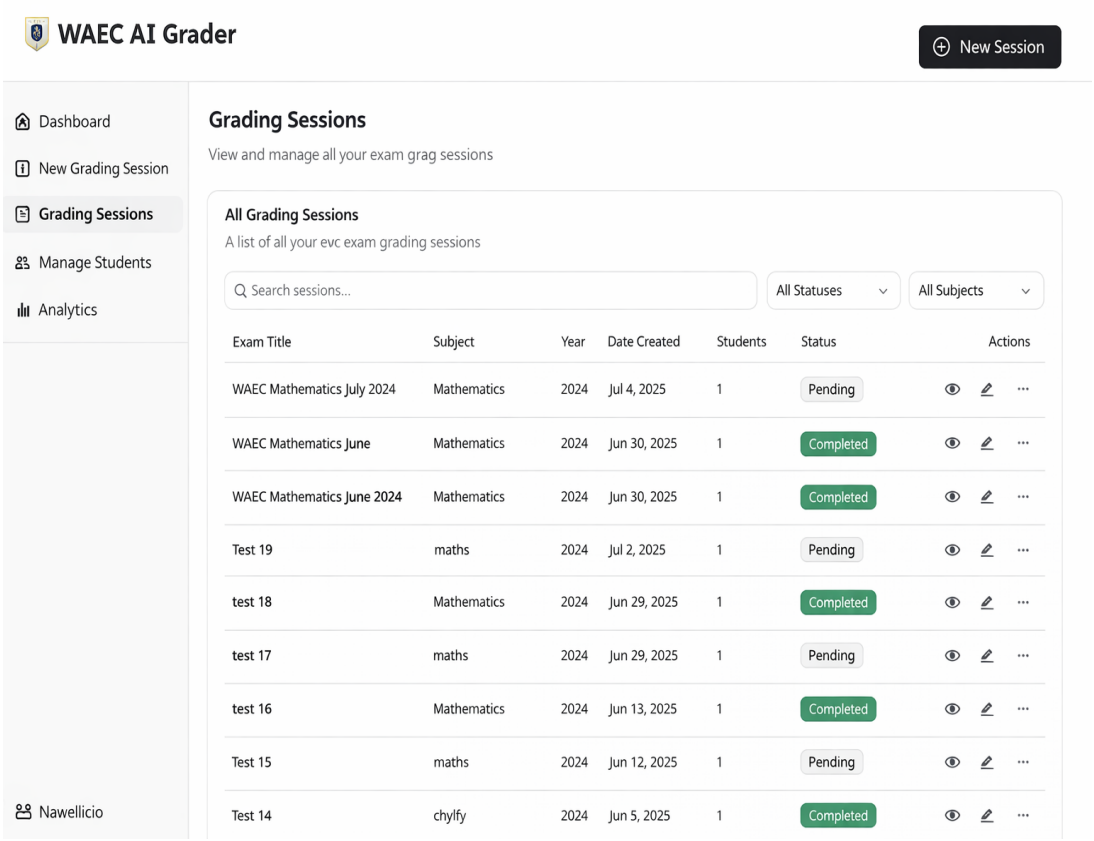

Fig.6 Image Showing the Grading Sessions Page

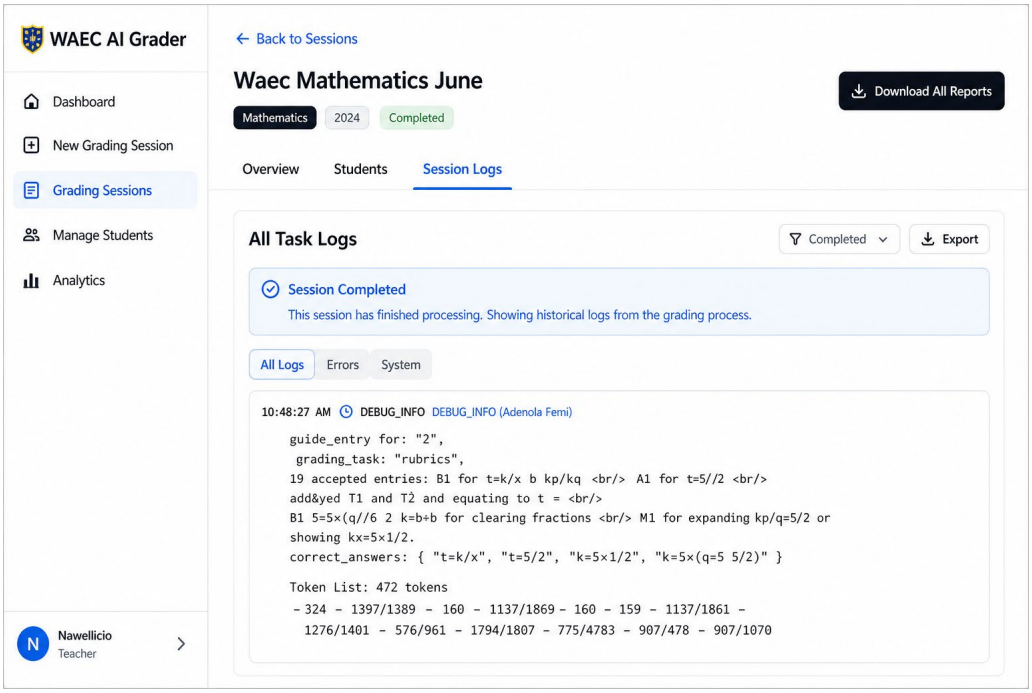

Fig.7 Image Showing Session Logs Obtained Via Web Sockets

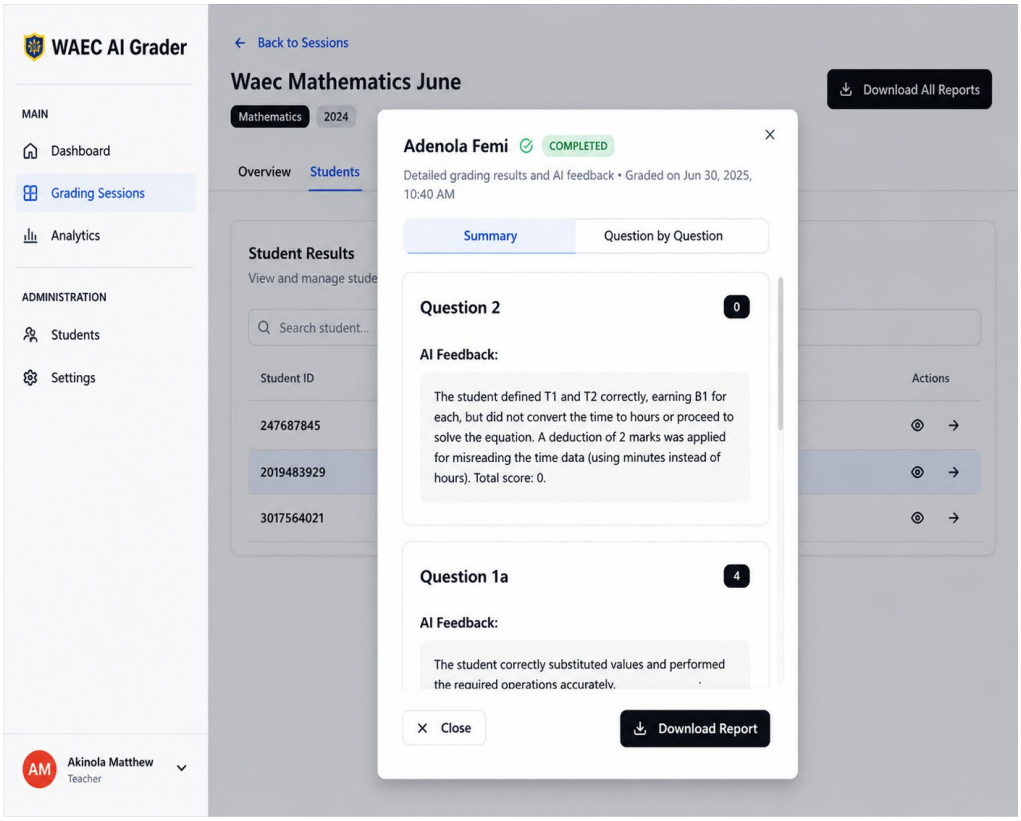

Fig.8 Image Showing the Results and Feedback Modal

The backend, developed with FastAPI, performs the following tasks:

1. *API Endpoints:* Receive grading requests and dispatch background tasks.
2. *WebSocket Integration:* Enable real-time status updates pushed to the frontend.
3. *Grading Orchestration:* Coordinate agent-based grading pipelines using LangGraph.

Celery workers are responsible for processing each uploaded exam paper. Task progress and logs are stored in Redis and streamed to the frontend via WebSockets.

A. Grading Agent Implementation

Currently, only the Mathematics Agent has been implemented. Its pipeline includes:

1. *Vision Parsing Agent:* A multimodal LLM (qwen/qwen2.5-vl-32b-instruct) parses the scanned exam paper, extracting questions and student answers.
2. *Grading Agent:* The deepseek/deepseek-r1-0528-qwen3-8b model grades answers based on a structured rubric containing the expected answer and a mark allocation guide.
3. *Aggregation:* Each question's score is compiled into a full report, which is sent back to the frontend.

This modular approach allows different agents to specialize in perception (OCR + structural parsing) versus reasoning (grading).

V. CONCLUSION

The implementation of an AI-powered theory grading system represents a significant step toward automating educational assessments, particularly in resource-constrained environments [22]. By leveraging LLMs within a modular architecture, the system achieves a combination of flexibility, efficiency, and real-time feedback. While the current version supports only mathematics scripts and relies on open-source/free-tier models—which introduces performance limitations—the modular design enabled by LangGraph allows future extension to additional subjects with minimal effort. Overall, the study validates the feasibility of automating theory grading using LLMs and agent-based design patterns. To improve the system, several enhancements are recommended. First, stronger vision–language models should be integrated to better handle variability in handwritten input. This may include commercial OCR solutions or fine-tuned multimodal models for improved parsing accuracy. Second, subject-specific grading agents should be developed for other theory-intensive disciplines such as Biology, Physics, Chemistry, and English to broaden the system's applicability. Third, to reduce latency and downtime, deploying models on dedicated servers or upgrading to premium API plans is advised. Additionally, incorporating advanced result analytics—such as performance trends, score distribution graphs, and detailed feedback reports—would enhance user experience. Finally, the system should be piloted in collaboration with educational institutions to evaluate real-world adoption, reliability, and scalability.

Declaration of Conflicting Interests
The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The authors received no financial support for the research, authorship, and/or publication of this article.

Use of Artificial Intelligence (AI)-Assisted Technology for Manuscript Preparation
The authors confirm that no AI-assisted technologies were used in the preparation or writing of the manuscript, and no images were altered using AI.

REFERENCES

[1] S. M. Darwish, R. A. Ali, and A. A. Elzoghabi, "An automated English essay scoring engine based on neutrosophic ontology for electronic education systems," *Appl. Sci.*, vol. 13, no. 15, art. 8601, Jul. 2023, doi: 10.3390/app13158601.

[2] I. Aggarwal, P. Gautam, and G. Parashar, "Automated subjective answer evaluation using machine learning," in *Proc. Int. Conf. Comput. Sci. (ICCS)*, 2023.

[3] M. Lundgren, "Large language models in student assessment: Comparing ChatGPT and human graders," *SSRN Electron. J.*, Jun. 24, 2024, doi: 10.2139/ssrn.4874359.

[4] S. Wang, T. Xu, H. Li *et al.*, "Large language models for education: A survey and outlook," *arXiv preprint* arXiv:2403.18105, Mar. 2024.

[5] S. Dikli and S. Bleyle, "Automated essay scoring feedback for second language writers: How does it compare to instructor feedback?" *Assess. Writing*, vol. 22, pp. 1–17, Oct. 2014, doi: 10.1016/j.asw.2014.03.006.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, … and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[7] J. Flodén, "Beyond human subjectivity and error: A novel AI grading system," *arXiv preprint* arXiv:2405.04323, 2024.

[8] A. Gobrecht, F. Tuma, M. Möller, T. Zöller, M. Zakhvatkin, A. Wuttig, H. Sommerfeldt, and S. Schütt, "Beyond human subjectivity and error: A novel AI grading system," *arXiv preprint* arXiv:2405.04323, May 2024.

[9] H. Xu, W. Gan, Z. Qi, J. Wu, and P. S. Yu, "Large language models for education: A survey," *arXiv preprint* arXiv:2405.13001, May 2024.

[10] A. O. Adeyanju, O. T. Oladele, and O. A. Adebayo, "Artificial intelligence-based essay grading system," *Int. J. Innov. Res. Multidiscip. Philos. Stud.*, vol. 2, no. 2, 2024.

[11] S. M. Darwish, R. A. Ali, and A. A. Elzoghabi, "An automated English essay scoring engine based on neutrosophic ontology for electronic education systems," *Appl. Sci.*, vol. 13, no. 15, art. 8601, 2023.

[12] F. Li, X. Xi, Z. Cui, D. Li, and W. Zeng, "Automatic essay scoring method based on multi-scale features," *Appl. Sci.*, vol. 13, no. 11, art. 6775, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/11/6775.

[13] S. M. Darwish, R. A. Ali, and A. A. Elzoghabi, "Automatic essay exam scoring system: A systematic literature review," *Appl. Sci.*, 2023.

[14] "Large language models and automated essay scoring of English language learner writing: Insights into validity and reliability," *J. Writing Anal.*, 2024.

[15] H. M. Alawadh, T. Meraj, L. Aldosari, and H. T. Rauf, "An efficient text-mining framework of automatic essay grading using discourse macrostructural and statistical lexical features," *SAGE Open*, vol. 14, no. 1, pp. 1–18, 2024.

[16] Y. Sun, J. Li, and H. Wu, "AI in teaching English writing: Automatic scoring and feedback system," *Appl. Math. Nonlinear Sci.*, vol. 9, no. 1, pp. 1203–1215, 2024.

[17] I. Gambo, F. Abegunde, O. Gambo, and R. Ogundokun, "GRAD-AI: An automated grading tool for code assessment and feedback in programming courses," *Educ. Inf. Technol.*, 2024.

[18] M. Messer, N. Brown, and M. Kölling, "Automated grading and feedback tools for programming education: A systematic review," *J. Comput. Assist. Learn.*, 2023.

[19] H. M. M. Ahmed and S. E. Sorour, "Classification-driven intelligent system for automated evaluation of higher education exam paper quality," *Educ. Inf. Technol.*, vol. 29, 2024.

[20] O. O. Akinwale and O. Tunde-Adeleke, "A structured dataset for automated grading: From raw data to processed dataset," *Data*, vol. 10, no. 6, art. 87, 2025.

[21] I. Eweoya, O. J. Adeniyi, A. O. Awoniyi, E. Mgbeahuruike, J. O. Adewuyi, T. O. Adigun, and Y. A. Mensah, "Design and implementation of a university attendance management system using geofencing," *Asian J. Comput. Sci. Technol.*, vol. 14, no. 1, pp. 12–18, 2025.

[22] H. G. Misgna *et al.*, "Artificial intelligence-based essay grading system," *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 11, 2023.