اونيۏرسيتي مليسيا ڤهڠ السلطان عبد الله
UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

FACULTY OF COMPUTING

SESSION 2025/2026

SEMESTER I

BCS3263

SOFTWARE QUALITY ASSURANCE

QUALITY DESIGN DOCUMENT

PROJECT

GRADELY (LEARNING PROGRESS AND PERFORMANCE PORTAL)

MODULE - MANAGE ASSIGNMENT

LECTURER:

PUAN ROSLINA BINTI MOHD SIDEK

STUDENT NAME & ID:

SYAZREEN ELYNA BINTI MUHAMMAD RAJAB

CB23140

TABLE OF CONTENT

ABSTRACT OF PROJECT

The Gradely Learning Progress and Performance Portal is a web-based educational system designed to enhance academic activities by centralizing assignment management, submission tracking, and student–lecturer interactions. At the core of this platform, the Manage Assignment module enables lecturers to create, organize, publish, and update assignments while providing students with a structured and accessible submission workflow. Given its importance in the teaching and learning process, ensuring the module's quality is essential. This Quality Design Document focuses on three ISO 25000 quality attributes which are Availability, Capacity, and Functional Appropriateness to guide the development of a dependable and user-centric module. These attributes ensure that assignment information remains consistently accessible, that the system can handle high submission loads and concurrent users, and that all features effectively support lecturers' and students' real academic needs. By evaluating these attributes through detailed scenarios and measurable indicators, this document establishes a strong quality framework that enhances performance, usability, and reliability across the module.

1.0 INTRODUCTION

The Gradely Learning Progress and Performance Portal is an educational web-based platform developed to support and enhance digital teaching and learning activities for both students and lecturers. The system integrates essential academic functionalities such as assignment management, submission tracking, feedback delivery, and performance visualization into a single, centralized platform. One of the core components of Gradely is the Manage Assignment module, which enables lecturers to create, organize, and publish assignments for students. Through this module, lecturers can set submission deadlines, upload instructional materials, manage assignment visibility through hide/unhide function, and monitor student submission activity. These functionalities support a structured academic workflow, ensuring that assignments are delivered clearly and promptly, and providing students and lecturers with a reliable and interactive environment for coursework management.

In accordance with the ISO/IEC 25010 model under the ISO 25000 SQuaRE series, this Quality Design document highlights three software quality attributes which are Availability, Capacity, and Functional Appropriateness that are fundamental to ensuring the reliability of the Manage Assignment module. ISO defines Availability as the degree to which a system is operational and accessible when required (ISO/IEC 25010, 2011), a critical requirement to ensure students and lecturers can consistently access assignment details such as visibility status and posted materials. Capacity, categorized under Performance Efficiency, refers to the system's ability to operate effectively within defined maximum system limits, including user load, number of assignment records, and file size thresholds (ISO/IEC 25010, 2011). Meanwhile, Functional Appropriateness is defined as the degree to which system functions support users in accomplishing their specific tasks efficiently (ISO/IEC 25010, 2011), making it highly relevant for ensuring that features such as assignment creation, editing, deadline configuration, and visibility management align with actual lecturer workflows. Together, these quality attributes form the foundation of a robust quality-oriented design for the Manage Assignment module, ensuring dependable operation, effective performance, and alignment with user needs.

2.0 BACKGROUND OF STUDY

Online learning platforms like Moodle, Google Classroom, and institutional systems have greatly improved the way assignments and assessments are distributed, but many still struggle to provide clear and continuous insights into student performance. Research shows that students often lack a structured view of their grades or performance trends across different assignments or modules, which limits their ability to self-reflect and improve (Zhang et al., 2024). Such performance transparency is a key quality for modern LMSs, as it supports learner motivation and ownership of learning (Koi-Akrofi et al., 2024).

Lecturers also face significant challenges in these existing systems. Many LMS platforms force instructors to rely on external tools such as spreadsheets or manual tracking to manage assignment records, track submissions, and give feedback. This fragmented workflow increases administrative burden, introduces risks of data inconsistency, and can delay feedback, which reduces teaching effectiveness (Fiqri, Alfarisy & Sutabri, 2023). Evaluations using ISO 25010 have identified performance inefficiency and usability issues in traditional LMSs, reinforcing the need for better-designed systems (Fiqri et al., 2023).

To address these shortcomings, the Gradely Learning Progress and Performance Portal is designed as a unified, quality-driven solution. The Manage Assignment module allows lecturers to publish assignments, set deadlines, upload materials, control visibility like the hide/unhide feature, and oversee submissions in a seamless environment. From a quality assurance design perspective, three ISO/IEC 25010 attributes are prioritized which are Availability, Capacity, and Functional Appropriateness (ISO/IEC 25010, 2011). High Availability ensures that both students and lecturers can reliably access assignment-related information around the clock, thereby reducing disruptions and supporting continuous engagement (IJERE, 2024). Capacity ensures the system can handle high loads such as large file uploads, multiple concurrent users, and many assignments without performance degradation, which is also highlighted by LMS quality evaluation models (Koi-Akrofi et al., 2024). Finally, Functional Appropriateness ensures that core functions like creating, editing, and managing assignments align with lecturers' real-world tasks and workflows, making the system meaningful and effective for academic use.

3.0 INTRODUCTION TO QUALITY ATTRIBUTES

In software engineering, quality attributes represent the non-functional characteristics of a system that determine its overall effectiveness, reliability, and user satisfaction. ISO/IEC 25010, part of the ISO 25000 SQuaRE series, provides a standardized framework for evaluating software product quality by defining key characteristics and sub-characteristics that are essential for assessing system performance, reliability, and suitability (ISO/IEC 25010, 2011). In the context of the Gradely Learning Progress and Performance Portal, the Manage Assignment module was designed with a focus on three critical quality attributes which are Availability, Capacity, and Functional Appropriateness. These attributes were chosen to ensure that the system consistently supports both lecturers and students in the assignment management process while maintaining performance efficiency and task relevance.

3.1 Availability

Availability is defined in ISO/IEC 25010 as the degree to which a system is operational and accessible when required (ISO/IEC 25010, 2011). In Gradely, availability ensures that assignment-related information including visibility status, uploaded materials, and deadlines is consistently accessible to both lecturers and students. Features such as the hide/unhide function directly support this attribute by allowing lecturers to control when assignments are visible, ensuring reliable access without system downtime or interruptions.

Key Aspects for Manage Assignment Module:

1. Continuous accessibility of assignments and deadlines.
2. Reliable hide/unhide functionality for controlled visibility.
3. Minimal downtime or service interruptions.

3.2 Capacity

Capacity, a sub-characteristic of Performance Efficiency in ISO/IEC 25010, refers to the system's ability to operate within defined maximum limits, including concurrent users, file sizes, and data records (ISO/IEC 25010, 2011). The Manage Assignment module must handle multiple assignments, large file uploads, and concurrent access by many students and lecturers.

Capacity ensures that the module maintains efficient performance under peak academic load, preventing delays or system slowdowns during critical periods such as submission deadlines.

Key Aspects for Manage Assignment Module:

1. Handling multiple simultaneous assignment uploads
2. Supporting large numbers of concurrent users
3. Managing large assignment file sizes without performance degradation

3.3 Functional Appropriateness

Functional Appropriateness measures the degree to which system functions support users in accomplishing their intended tasks effectively (ISO/IEC 25010, 2011). This attribute ensures that lecturers can efficiently perform key tasks, such as creating assignments, editing content, updating deadlines, and monitoring student submissions. A function is considered appropriate when it contributes directly to the intended task without unnecessary complexity or redundant steps. By emphasizing functional appropriateness, the module enhances usability, reduces task friction, and ensures that users can perform essential operations accurately and efficiently.

Key Aspects for Manage Assignment Module:

1. Assignment creation and editing that aligns with lecturer workflows.
2. Accurate deadline configuration and updates.
3. Clear visibility and access to assignment details and downloadable resources.

4.0 QUALITY ATTRIBUTES SCENARIO IN YOUR PROJECT

4.1 Availability

4.1.1 Scenario 1: Lecturer Edits Assignment Instructions During Active Hours

Lecturers often revise instructions after publishing assignments. The system must remain continuously available during these updates to prevent disruption, especially during daytime peak usage. High availability ensures uninterrupted editing and aligns with ISO/IEC 25010, which emphasizes access reliability and operational readiness.

Source of Stimulus	Lecturer
Stimulus	Attempts to edit assignment instructions.
Artifact	• Assignment Editing Interface • Assignment Database Service • Backend for update operations
Environment	Normal working hours with moderate traffic.
Response	The system retrieves the assignment data, displays the editing form, saves updated information, and reflects changes without interrupting access for other users.
Response Measure	Page loads < 2 seconds, 99% uptime during editing, and zero update failures.

4.1.2 Scenario 2: Student Accesses Assignment Minutes Before Deadline

Students may frequently open the assignment page right before submission closes. The system must remain available to prevent access errors that could unfairly impact submission opportunities. This is especially critical during high-traffic final minutes, where system downtime would severely affect student performance and system trustworthiness.

Source of Stimulus	Student
Stimulus	Student opens the assignment page minutes before the submission deadline.

Artifact	• Assignment Viewing Interface • Content Delivery System • Deadline Retrieval Function
Environment	Peak submission time with high concurrency logins and page visits by many students.
Response	System successfully loads the assignment details, displays instructions and deadlines, and allows access to submission functionality without delay or system crash.
Response Measure	100% task success rate, response time < 3 seconds even under peak load, and zero failed requests.

4.1.3 Scenario 3: Lecturer Unhides an Assignment Before Class Starts

Lecturers may prepare assignments in advance and unhide them moments before a class session. The system must be available at the exact moment the lecturer performs this action to avoid delays in student access, ensuring smooth teaching workflows.

Source of Stimulus	Lecturer
Stimulus	Lecturer selects “Unhide Assignment” to make the assignment visible to students immediately.
Artifact	Assignment visibility control function
Environment	Typically, 5-10 minutes before a class session, and low-to-moderate traffic.
Response	The system updates visibility status instantly and broadcasts changes across all student accounts without delay.
Response Measure	Visibility update occurs within 3 second, 100% propagation accuracy, and no system delay or data mismatch.

4.2 Capacity

4.2.1 Scenario 1: Multiple Students Uploading Assignments Simultaneously

During assignment deadlines, it is normal for many students to upload their files at the same time. This scenario tests whether the Gradely system can reliably accept and process simultaneous upload requests without slowing down, timing out, or causing corrupted uploads. Even though the system enforces a 20MB file-size limit, simultaneous uploads still generate high server demand on storage I/O, bandwidth, and request handling. The system should queue or parallel-process these uploads while maintaining stable performance.

Source of Stimulus	Student
Stimulus	50 students simultaneously upload PDF/Docx files (each up to the max limit of 20MB).
Artifact	• The file upload function • Backend storage handler
Environment	Peak usage period near an assignment submission deadline, high server load.
Response	The system processes all upload requests concurrently, queues excess requests if needed, and completes uploads without crashing.
Response Measure	95% of uploads complete within under 5 seconds and no failed or corrupted uploads occur under simultaneous load.

4.2.2 Scenario 2: Large Number of Concurrent Users Accessing Assignment Page

This scenario focuses on system capacity when many users access the assignment page simultaneously, such as during class announcements or when a lecturer publishes new instructions. An LMS must support many concurrent read requests without performance degradation. Because read operations occur more frequently than file uploads, efficient caching and optimized database queries are crucial.

Source of Stimulus	Student
Stimulus	50 concurrent users loading the assignment page within the same 10-second interval.
Artifact	• Frontend assignment module • Backend for assignment retrieval • Database query logic
Environment	Normal academic hours where high concurrency is expected across multiple classes.
Response	The system retrieves and serves assignment details smoothly without slow rendering or timeouts.
Response Measure	Page load time remains under 2 seconds for 98% of users.

4.2.3 Scenario 3: Uploading Large Assignment Files Near System Size Limit (20MB)

Even with a file size limit of 20MB, some students will still upload large video clips, high-resolution PDFs, or documents containing images. Large-file handling must not degrade performance for other users. The system should validate file size before upload, stream the file efficiently, and maintain responsiveness. Resource spikes (CPU, memory, I/O) must remain within acceptable boundaries.

Source of Stimulus	Student
Stimulus	A single upload of a 19.8MB file requiring validation and storage.
Artifact	• File upload validation logic • Backend storage system
Environment	Heavy course activity where many assignments and objects are being accessed.
Response	The system validates the file size, streams the upload without freezing the UI, and stores the file efficiently.
Response Measure	Upload completes within 8 seconds, server CPU usage does not exceed 70%, and no other user experiences noticeable slowdown.

4.3 Functional Appropriateness

4.3.1 Scenario 1: Lecturer Creates a New Assignment with Detailed Requirements

When lecturers create assignments, the system must only provide relevant and essential fields that directly support the task. A functionally appropriate system ensures that the process is intuitive, avoids unnecessary fields, prevents redundant steps, and supports important academic components such as deadlines, descriptions, and attachments. This scenario ensures that assignment creation aligns with lecturer workflows and reduces the likelihood of confusion or errors.

Source of Stimulus	Lecturer
Stimulus	Lecturer fills in assignment title, instructions, deadline, marks, and uploads materials.
Artifact	• Assignment creation form • Data Validation Engine • Assignment Management Controller
Environment	Normal usage, lecturer interacting with a structured UI designed for efficient data entry.
Response	System validates inputs, saves the assignment, uploads materials, and displays confirmation without requiring unnecessary fields or extra steps.
Response Measure	Task completed within 2–3 minutes, mandatory fields are only those relevant to assignment creation, confirmation within 1 second.

4.3.2 Scenario 2: Lecturer Updates Deadline to Extend Submission Period

Lecturers often update deadlines due to changes in teaching schedules or student needs. Functional appropriateness ensures that the deadline-editing feature is enhanced, requires minimal action, and avoids unnecessary re-entry of unrelated assignment data. The system must support fast and intuitive deadline modification and ensure that changes are reflected instantly across all student interfaces.

Source of Stimulus	Lecturer
Stimulus	Lecturer clicks “Edit Deadline” and selects a new closing date and time.
Artifact	Deadline modification function
Environment	Normal usage, students may be online but the update should not interrupt ongoing access.
Response	System validates the new deadline, updates the assignment metadata, and immediately reflects the changes for all students without requiring re-creation of the assignment.
Response Measure	Deadline change saved within < 2 second, UI refresh within 2 second, 100% accurate deadline visibility.

4.3.3 Scenario 3: Student Views Assignment Details and Downloads Resources

Students require clear and accurate information to complete assignments. Functional appropriateness ensures the module provides complete and correct details, avoids unnecessary complexity, and supports essential user tasks such as reading instructions and downloading files.

Source of Stimulus	Student
Stimulus	Student opens the assignment page and selects files to download.
Artifact	• Assignment viewing interface • File Access Handler
Environment	Normal-to-moderate usage, multiple students may be accessing the same assignment.
Response	System accurately displays assignment instructions, and deadlines in a structured and clear layout where the file downloads function work.
Response Measure	Page loads within 2 seconds, download links functional 100% of the time, no missing or incorrect information.

5.0 TACTICS FOR EACH ATTRIBUTE

5.1 Availability

5.1.1 Scenario 1: Lecturer Edits Assignment Instructions During Active Hours

Stimulus	Lecturer attempts to edit assignment instructions during normal working hours.
Tactics to Control Response	**1. Record Locking:** Lock only the assignment being edited using Laravel database transactions to prevent conflicts while allowing other assignments to be accessed. **2. Auto-Save Drafts:** Use periodic saving of the edited content in the database as a draft to prevent data loss. **3. Error Handling with Notifications:** Catch any save errors and display user-friendly messages so the lecturer knows the system is retrying automatically.
Response	Assignment edits are saved without interrupting other users' access and page responds within <2 seconds, ensuring minimal disruption.

These tactics ensure the system remains highly available while a lecturer edits an assignment. Record locking allows safe concurrent use, auto-save protects against crashes or accidental navigation, and error handling ensures any interruptions are immediately addressed. Combined, they maintain continuous accessibility without impacting other users.

5.1.2 Scenario 2: Student Accesses Assignment Minutes Before Deadline

Stimulus	Student opens assignment page right before submission deadline.
Tactics to Control Response	**1. Page Caching:** Store assignment page content or database query results temporarily to reduce repeated MySQL queries and speed up loading.

	2. **Asynchronous Content Loading (AJAX):** Load critical elements like submission buttons and instructions asynchronously so the page becomes usable immediately. 3. **Lightweight UI Rendering:** Only render essential elements (instructions, deadlines, submission links) to reduce load time.
Response	Students access assignment pages without delay, page loads <3 seconds, and submission functionality fully available even under high traffic.

These tactics improve system availability during peak usage. Page caching reduces server/database load, asynchronous loading allows students to interact with the page immediately, and lightweight UI ensures the page renders quickly. Together, they prevent downtime or failed submissions even during high-stress periods.

5.1.3 Scenario 3: Lecturer Unhides an Assignment Before Class Starts

Stimulus	Lecturer clicks "Unhide Assignment" to make it visible immediately.
Tactics to Control Response	1. **Real-time Visibility Update:** Update assignment status in MySQL immediately and propagate changes to all users. 2. **In-Memory Status Flags:** Store temporary visibility state to prevent repeated database queries during rapid updates. 3. **Notification Trigger:** Automatically notify students of new visibility so they can access the assignment immediately.
Response	Assignment becomes visible within 3 seconds, and all students see updated status instantly.

These tactics ensure immediate and reliable assignment visibility. Real-time updates guarantee accuracy, in-memory flags prevent database bottlenecks, and notifications improve student awareness. Combined, the system remains responsive and dependable for all users.

5.2 Capacity

5.2.1 Scenario 1: Multiple Students Uploading Assignments Simultaneously

Stimulus	50 students upload files up to 20MB at the same time.
Tactics to Control Response	**1. Parallel Upload Processing (Laravel Jobs):** Use Laravel queues/jobs to handle multiple uploads concurrently. **2. File Size Validation:** Validate file size before upload to prevent server overload. **3. Temporary Queuing:** Queue excess uploads to process sequentially if resources are temporarily saturated.
Response	All uploads complete within 5 seconds, no files are lost or corrupted, and system remains responsive.

These tactics maintain capacity during high-load periods. Parallel processing ensures multiple uploads are handled simultaneously, file validation prevents oversized files from overloading the server, and temporary queuing maintains orderly processing. Together, the system can handle submission peaks effectively.

5.2.2 Scenario 2: Large Number of Concurrent Users Accessing Assignment Page

Stimulus	50 users access assignment page within 10 seconds.
Tactics to Control Response	**1. Query Caching (Laravel Cache):** Cache frequently accessed assignment data to reduce MySQL load. **2. Connection Pooling:** Reuse database connections efficiently to reduce overhead from creating new connections. **3. Lightweight Page Rendering:** Render only essential content for the initial page load to reduce processing and response time.

Response	Page loads in <2 seconds for 98% of users, and system maintains stable performance.

These tactics optimize capacity under high concurrency. Query caching reduces repeated database access, connection pooling saves server resources, and lightweight rendering speeds up response. Combined, they allow many students to access the page simultaneously without performance degradation.

5.2.3 Scenario 3: Uploading Large Assignment Files Near System Size Limit (20MB)

Stimulus	Student uploads a 19.8MB file.
Tactics to Control Response	**1. Chunked File Upload:** Upload large files in smaller segments to reduce memory consumption. **2. Pre-upload Validation:** Check file size and type before uploading to prevent unnecessary processing. **3. Progress Feedback:** Show upload progress to the student so they know the system is working.
Response	Upload completes within 5 seconds, CPU usage <70%, and system remains responsive for other users.

These tactics ensure that large file uploads do not overload the system. Chunked upload manages memory efficiently, pre-upload validation prevents resource waste, and progress feedback improves user experience. Combined, students can upload large files safely and reliably.

5.3 Functional Appropriateness

5.3.1 Scenario 1: Lecturer Creates a New Assignment with Detailed Requirements

Stimulus	Lecturer fills in assignment title, instructions, deadline, marks, and uploads materials.
Tactics to Control Response	**1. Guided Form Input:** Only include fields required for assignment creation to reduce complexity. **2. Real-time Input Validation:** Check for missing or incorrect information while the lecturer fills the form. **3. Confirmation Feedback:** Display a success message when the assignment is saved.
Response	Assignment is created in 2–3 minutes; confirmation shown within 1 second; only relevant fields required.

These tactics ensure assignment creation is user-friendly and accurate. Guided input reduces unnecessary steps, real-time validation prevents errors, and confirmation feedback reassures the lecturer. Together, they make assignment creation efficient and reliable.

5.3.2 Scenario 2: Lecturer Updates Deadline to Extend Submission Period

Stimulus	Lecturer clicks “Edit Deadline” and selects a new date/time.
Tactics to Control Response	**1. Instant Deadline Update:** Immediately update the assignment deadline in the database. **2. Backend Metadata Update:** Ensure the new deadline is saved reliably in MySQL. **3. Student Notification:** Inform students of the new deadline to prevent confusion.
Response	Deadline updated in <2 seconds, all students see new deadline immediately, and UI refreshes instantly.

These tactics guarantee functional appropriateness by ensuring deadlines are updated accurately and quickly. Instant updates reflect changes immediately, metadata update preserves data integrity, and notifications keep students informed. Together, they maintain accurate workflows.

5.3.3 Scenario 3: Student Views Assignment Details and Downloads Resources

Stimulus	Student opens assignment page and downloads resources.
Tactics to Control Response	**1. Optimized Data Retrieval:** Fetch assignment data efficiently using MySQL queries. **2. Cached Downloads:** Store frequently downloaded files to reduce load on storage and database. **3. Access Validation:** Ensure only authorized students can download files to prevent errors.
Response	Page loads <2 seconds, all downloads work 100% of the time, and information is accurate and complete.

These tactics ensure functional appropriateness for students accessing assignments. Optimized retrieval improves speed, cached downloads reduce repeated storage/database load, and access validation prevents unauthorized access. Combined, students can view and download assignments reliably and efficiently.

6.0 CONCLUSION

The Quality Design Document (QDD) for the Gradely Learning Progress and Performance Portal, focusing on the Manage Assignment module, serves as a blueprint for implementing Software Quality Assurance (SQA) practices throughout the system's development. By identifying critical software quality attributes such as Availability, Capacity, and Functional Appropriateness and analyzing real-world scenarios, the QDD ensures that SQA principles are embedded in the design phase. Scenarios such as lecturers editing assignments during active hours, students accessing assignments near deadlines, and multiple simultaneous file uploads illustrate potential quality risks and provide a basis for defining measurable system behavior, which is central to SQA activities.

The tactics defined in this document, including record locking, auto-saving drafts, page caching, parallel upload processing, chunked uploads, optimized queries, guided form inputs, real-time validation, and student notifications, directly support SQA by translating these quality attributes into actionable design decisions. Implementing these tactics allows the system to maintain reliable performance, prevent errors, and ensure functional correctness during assignment creation, management, access, and submission. Overall, the QDD and its associated tactics provide a structured SQA approach, ensuring that the Manage Assignment module is robust, maintainable, and user-centric, supporting both lecturers' and students' academic workflows effectively.

7.0 REFERENCES

ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models, ISO, Mar. 2011. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?start=5

G. Y. Koi-Akrofi, H. Tanye, S. C. Quist, J. Koi-Akrofi, E. Gaisie, and M. Agangiba, "Towards a Software Quality Factor Assessment Model for Learning Management Systems (LMS)," Indian Journal of Science and Technology, vol. 17, no. 23, pp. 2463–2468, Jun. 2024. DOI: 10.17485/IJST/v17i23.1006.

A. M. Fiqri, A. Alfarisy, and T. Sutabri, "Evaluasi Kualitas Learning Management System berdasarkan ISO 25010 pada SMK Muhammadiyah 1 Palembang," Explore: Jurnal Sistem Informasi dan Telematika, vol. 14, no. 1, 2023. DOI: 10.36448/jsit.v14i1.3116.

X. Zhang, V. C. S. Lee, D. Xu, J. Chen, and M. S. Obaidat, "An Effective Learning Management System for Revealing Student Performance Attributes," arXiv preprint abs/2403.13822, Mar. 2024.

H. Elmunsyah, A. Nafalski, A. P. Wibawa, and F. A. Dwiyanto, "Understanding the Impact of a Learning Management System Using a Novel Modified DeLone and McLean Model," Education Sciences, vol. 13, no. 3, article 235, 2023, doi:10.3390/educsci13030235.

N. A. Ahmad, A. A. Mayouf, N. F. Elias, and H. Mohamed, "Learning management system instrument development based on Aiken's V technique," International Journal of Evaluation and Research in Education (IJERE), vol. 13, no. 5, pp. 3211–3219, Oct. 2024, doi: 10.11591/ijere.v13i5.28925.