Science World Journal Vol. 21(No 2) 2026 https://dx.doi.org/10.4314/swj.v21i2.26
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

VERITY: DEVELOPMENT OF AN INTEGRATED MULTI-ROLE ONLINE EXAMINATION PLATFORM WITH AI-BASED PROCTORING, PLAGIARISM DETECTION, AND LLM-CONTENT ANALYSIS

*A.O. Ogar, S. Muhammed, A.S. Nur, F.O. Muhammed

Department of Computer Science, Faculty of Computing, Nile University of Nigeria

*Corresponding Author Email Address: austinolomogar@gmail.com

ABSTRACT
The rapid growth of online examinations has highlighted a rather disjointed problem in academic integrity enforcement: proctoring systems, plagiarism detectors, and large language model (LLM) content detectors are each deployed as stand-alone tools, leaving institutional reviewers to reconcile evidence across disconnected dashboards. This paper presents Verity, an integrated multi-role online examination platform that combines AI-based proctoring, multi-layer plagiarism detection, and LLM-based content analysis in a single architecture. The system has three role-separated user interfaces, student, proctor, and administrator, tied to a common Unified Student-Examination Record (USER) data model. Verity was developed using the Design Science Research methodology and assessed through a 28-participant System Usability Scale (SUS) study and a three-expert heuristic review. The findings reveal an average SUS score of 80.1 (Good) with no participant scoring below 70 and no major usability violations remaining to be fixed. This research offers a replicable architecture and a prototype that leads the way in the design of trustworthy and scalable online assessments.

Keywords: *Online examination, AI proctoring, plagiarism detection, LLM-content detection, academic integrity, design science research.*

INTRODUCTION
The move to online assessment, which was accelerated by the COVID-19 pandemic, has fundamentally changed how higher education examinations are conducted (Topuz, 2024). Online exams can offer scalability and a variety of time slots for students to take the test that physical testing cannot match. However, they have also fed a burgeoning academic integrity crisis. Studies have found that rates of academic misconduct are almost always higher in unsupervised online settings than in settings with exam invigilation (Sozon *et al.*, 2024). Besides, the availability of large language models (LLMs) like ChatGPT to the public has become a concern, as students can use them to create well-structured responses that can bypass even the most sophisticated similarity-based plagiarism detectors (Evangelista, 2025; Balalle & Pannilage, 2025). As the validity of a student's submission is the very foundation of the value of an institution's credentials, this breakdown has implications far beyond the introduction of new technologies.
Considerable commercial and research activities have focused on one or another aspect of the problem. AI proctoring tools like ProctorU, Respondus Monitor and Honorlock utilise webcams, track gaze and detect behavioural anomalies (Flores Zavaleta, 2024). Turnitin and iThenticate plagiarism checkers detect copied text from the Internet or the institution's documents (Ogwueleka, 2025). Recently, the emergence of LLM-content detectors such as Originality.ai and GPTZero has been reported; these tools recognize AI-generated writing (Sozon *et al.*, 2026). Systematic reviews show that each type, when used individually, can generate value (Han et al., 2024; Malhotra & Chhabra, 2026). However, these approaches are by their very nature still disjointed. One would need to buy and integrate multiple distinct platforms even within a single institution. Some of these platforms would probably still have different dashboards and student-facing flows. Not only does this type of fragmentation lead to extra administrative work, but it also causes integrity blind spots: a student who behaves in a manner that does not trigger a proctoring alert may still turn in a plagiarized or AI-generated paper without the availability of a single tool that the same reviewer can use to see both the signals.
This means that preserving the integrity of online testing cannot be achieved by simply identifying a single perfect remedy for a single problem. It is more of a succession of different hurdles. Only by the complete fusion of these various elements can meaningful and efficient implementation of the law be possible. Reviewers get the richest picture of integrity when data from proctoring, plagiarism detection, and LLM-content scoring are combined into a single data model and presented through interfaces suitable for different roles. Such an in-depth picture is something no single tool by itself could accomplish.
To address this deficiency, we introduce Verity, a comprehensive multi-role online examination platform that integrates AI-based proctoring, multi-level plagiarism detection, and LLM-based content analysis. Verity operates through three role-targeted interfaces: a student examination flow that is supported by identification verification, system checks, and assessment delivery in mixed formats; a proctor dashboard for real-time monitoring with anomaly alerting and flag review; and an administrator panel with plagiarism and AI-content reporting capabilities. The system has been designed using the Design Science Research methodology (Hevner et al., 2004; Peffers et al., 2007) and evaluated using the System Usability Scale (SUS) (Brooke, 1996) and Nielsen's heuristic evaluation method (1994).

This paper presents the following contributions: (i) a framework that unites online examination integrity components in one system, which include live proctoring, plagiarism check, and LLM-generated content detection; (ii) a multi-role interface design based on role-specific task analysis; (iii) a working high-fidelity prototype

Science World Journal Vol. 21(No 2) 2026 https://dx.doi.org/10.4314/swj.v21i2.26
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

that shows the system architecture; and (iv) validated design SUS and heuristic evaluation results. The rest of the paper is structured as follows: Section 2.0 provides an account of the materials and methods employed in this study; Section 3.0 outlines the SUS study, expert heuristic evaluation, and functional verification results; and Section 4.0 reflects on the implications, compares with existing tools, discusses limitations, and outlines future work directions.

MATERIALS AND METHODS

Research Approach

The research in the paper is based on the Design Science Research (DSR) framework by Hevner et al. (2004) and Peffers et al. (2007). If the main research result is a manually created artifact designed to address a well-identified real-world problem and expected to demonstrate its usefulness and originality, then DSR is the preferred method. The DSR method comprises three partially overlapping cycles: the Relevance Cycle (problem formulation in the context), the Design Cycle (creating and enhancing the artifact), and the Rigor Cycle (research based on prior knowledge and validation of the artifact against usefulness criteria).

Problem Formalisation

Let $S = \{s_1, \ldots, s_n\}$ denote a set of students taking an online examination. Each student s_i produces a submission X_i across question types Q = {MCQ, essay, code}. An integrity assessment system must evaluate s_i across three dimensions: proctoring integrity P_i (conformance of behavioural signals to permitted conduct over face presence, gaze direction, tab switches, audio anomalies, and multiple-person detection); plagiarism integrity L_i (proportion of X_i matching an external corpus C above a similarity threshold $\tau = 0.20$, consistent with Turnitin industry practice); and AI-content integrity A_i (the estimated probability $P(AI \mid r_{ij})$ that an LLM generated response r_{ij}). A fully integrated assessment for student s_i is therefore the tuple Integrity(s_i) = $\langle P_i, L_i, A_i \rangle$. The problem addressed is the absence of any existing system that computes and surfaces this full tuple through role-appropriate interfaces. Success is defined as a deployed prototype that captures all three dimensions, presents them through role-specific interfaces, and achieves a mean SUS score ≥ 70 across all three roles.

Baseline Analysis

Three dominant baseline architectures were characterised in the systematic reviews of Han *et al.* (2024) and Malhotra & Chhabra (2026). Standalone proctoring systems (e.g., ProctorU, Respondus Monitor) operate as browser lockdown clients with server-side webcam stream analysis; they offer no integration with submitted content. Document similarity plagiarism detectors (e.g., Turnitin, iThenticate) accept post-submission uploads and return batch similarity reports; they lack real-time capability and integration with proctoring signals. LLM-content detectors (e.g., Originality.ai, GPTZero) accept text and return AI probability scores in isolation, fully detached from examination context. The architectural gap shared by all three is the absence of a shared data model that binds proctoring events, submission content, plagiarism scores, and AI-content probabilities to a single student-examination record, accessible in real time to differentiated user roles.

System Architecture

Verity is a three-tier, multi-role web application with a shared services layer (Figure 1). The presentation layer renders three role-separated interfaces; the routing layer enforces role-based access control (RBAC); the core services layer hosts the four integrity engines (Identity Verifier, Proctoring Engine, Plagiarism Engine, AI-Content Detector); the data model layer enforces the Unified Student-Examination Record (USER) schema; and the persistence layer stores submissions and event logs. The presentation layer never accesses persistence directly; the core services layer mediates all access.

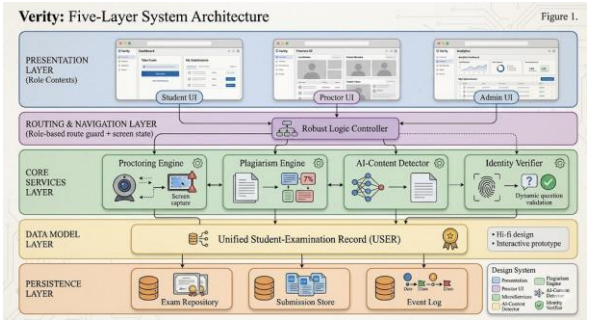

Figure 1: Verity five-layer system architecture.

A deliberate goal of this architecture is replicability: the five layers are decoupled and technology-neutral, so the platform is intended to serve as a reusable blueprint rather than a single proprietary product. Because the presentation, routing, core-services, data-model, and persistence layers communicate only through the Unified Student-Examination Record (USER) schema and well-defined service interfaces, an implementer can substitute any individual engine — for example, replacing the simulated face comparator with a locally hosted FaceNet model or the AI-content classifier with a fine-tuned RoBERTa model — without altering the surrounding layers. The component specification, role-permission matrix (Table I), proctoring-signal thresholds (Table II), and USER schema reported in this paper are documented in full precisely so that an independent team can re-implement Verity on an open, vendor-neutral stack; the reference prototype and its design assets are made available by the authors to support open replication and extension.

Unified Student-Examination Record (USER)

The USER data model is the keystone of the architecture. Each student-examination session is a persistent record containing student and exam identifiers, identity verification status, system-check results, an array of proctor events, an array of responses, plagiarism scores, AI-content scores, and the integrity tuple $\langle P_i, L_i, A_i \rangle$. All four core services write to and read from this single record, so cross-dimensional analysis (for example, ranking students whose live behaviour is clean but whose essay scores carry a high AI probability) becomes a single query rather than a manual reconciliation across vendor reports.

Role-Separated Interface Architecture

Verity realizes three independent front-end contexts that share the same backend API, each showcasing only the data and controls relevant to its particular role. Students can only view their own exam status and results; proctors can view live proctor events across all active students, mark severity, and take notes, but they cannot view raw response content; administrators have full post-

Science World Journal Vol. 21(No 2) 2026 https://dx.doi.org/10.4314/swj.v21i2.26
www.scienceworldjournal.org
ISSN: 1597-6343 (Online), ISSN: 2756-391X (Print)
Published by Faculty of Science, Kaduna State University

examination read access and write access to integrity rulings. This separation corresponds to the principle of least privilege (Saltzer & Schroeder, 1975) and prevents role cross-contamination. Table I gives an overview of role permissions.

Table I: Role Permission Matrix

Capability	Student	Proctor	Admin
View your exam state.	Read	—	Read
Submit responses	Write	—	—
View live proctor events.	—	Read	Read
Flag severity/notes	—	Write	Write
Raw response content	Own only	—	Read
Plagiarism / AI evidence	Summary	—	Full
Adjudication ruling	—	—	Write
Analytics dashboards	—	—	Read

Student Interface

The student's pathway is the progressive, gated sequence of five screens. S1, Dashboard shows the list of exams with a contextual main button whose status depends on the completion of system checks and exam window availability. S2, Identity Verification consists of three consecutive sub-steps: government ID upload (JPEG/PNG/PDF ≤ 5 MB), facial biometric capture using browser MediaDevices API, face-to-ID match confirmation. If a match confidence ≥ 80%, set identity_verified = TRUE; if fail is continuous, route to proctor review. S3, Pre-Examination System Check is conducted with 5 automated checks (camera, microphone, browser API support, upload bandwidth ≥ 1 Mbps, fullscreen capability); you must pass all of them before the "Join exam" button is enabled.

S4, Live Examination comes with a three-panel layout with a question navigator, an active-question panel, and an exam-status bar. MCQ items autosave on selection. Essay items show a live Originality Indicator as a colour-coded ring that updates every 30 seconds while the student types, a real-time deterrence signal not present in existing commercial tools. Code items are created using a Monaco-style editor and a "Run tests" feature that runs against hidden test cases. Exiting out of fullscreen mode logs a TAB_SWITCH event to proctor_events[]. S5, the Results & Integrity Report, not only provides the student with their scores but also their personal $\langle P_i, L_i, A_i \rangle$ in visuals to represent the levels, but the detailed evidence that supports these levels is kept back for the admin role to avoid reverse-engineering of detection thresholds.

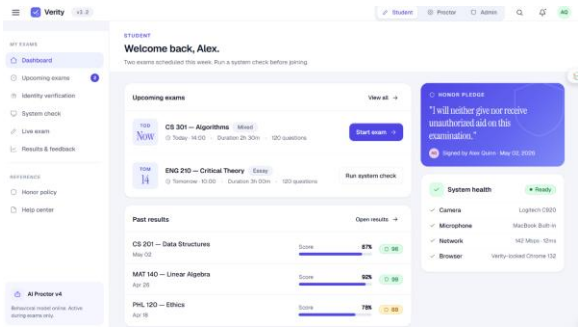

Figure 2: Student interface — Dashboard, Identity Verification, System Check, Live Examination, and Results screens.

Proctor Interface

The proctor interface consists of three screens. P1 Live Monitoring Dashboard displays students on exam live as a grid showing their webcam images at three zoom levels (Overview = 30 tiles, Standard = 12, Focus = 4) with filter options for alert levels, exam name, and student name. Each tile features a face detection bounding box, a live recording icon, and an alert count badge coloured according to the highest alert level. P2 Flag Review Queue displays the unresolved integrity flags in a way that is both sortable and supports bulk actions for low-severity items. P3 Student Detail Overlay reveals a split-screen view showing the webcam and screen-share video streams on the left, the complete incident timeline on the right. It supports note-taking, escalation, and severity resolution. All actions are recorded in the USER record, along with a proctor identifier and timestamp, for audit purposes.

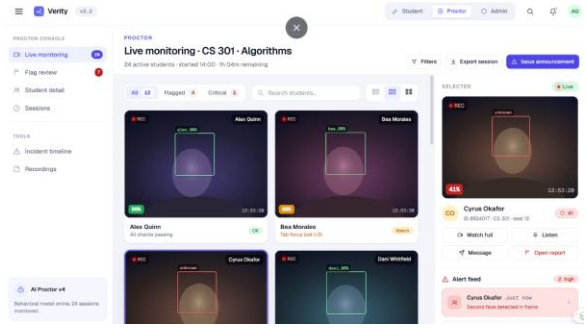

Figure 3: Proctor interface — Live Monitoring Dashboard, Flag Review Queue, and Student Detail overlay.

Administrator Interface

The administrator interface has three different screens. A1, Institution Overview shows institution-wide examination performance metrics (active examinations, students in session, open flags, mean integrity score) and a table of recent examinations. A2, Plagiarism & AI-Content Report is Verity's most unique screen. A dual-panel layout shows the full essay response on the left with three simultaneous, colour-coded highlight types, orange for web-source matches, purple for peer-submission matches, and blue for AI-generated regions, and an evidence summary on the right showing the three integrity gauges, ranked top-matched sources, and a formal adjudication control (No Violation / Warning / Academic Misconduct) that needs a written justification. Simultaneous multi-type highlighting in a single

reading view is the key design feature that most clearly differentiates Verity from existing tools, which require reviewers to context-switch between separate reports. A3, Analytics Panel represents the score against integrity composite as a scatter and a trend chart of mean integrity scores across examination sittings, thereby supporting institutional quality management.

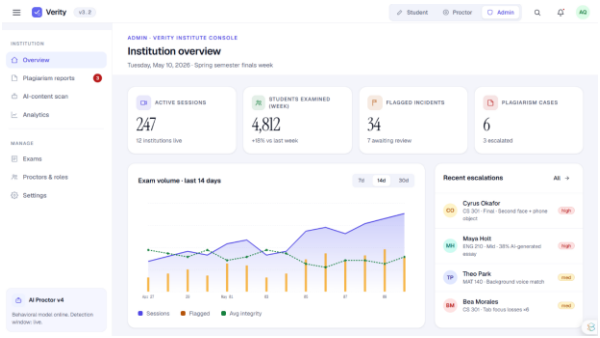

Figure 4: Administrator interface — Institution Overview, Plagiarism & AI-Content Report, and Analytics Panel.

Proctoring Signal Thresholds
The proctoring engine assesses six signals as summarised in Table II.

Table II: Proctoring Signal Thresholds

Signal	Alert Trigger	Severity
Face absence	> 5 s without face detected	Medium
Multiple faces	≥ 2 faces in frame	High
Gaze off-screen	> 10 s sustained	Low
Tab switch	Fullscreen exit/focus loss	Medium
Audio anomaly	Voice/keyboard burst > threshold	Low
Identity drift	Face mismatch confidence < 60%	High

Plagiarism detection is triggered when $\tau = 0.20$ (a very serious case at $\tau \geq 0.40$). We derive the probability P(AI | response) from the features' perplexity and burstiness, as explained in Gehrmann, Strobelt & Rush (2019). If this probability exceeds 0.75, the text is marked as AI; if it exceeds 0.90, it is marked as high severity.

Implementation — Technology Stack
Verity is a standalone HTML5 combo that uses vanilla JavaScript (ES2022) along with CSS3 custom properties to manage design tokens and themes. For camera capture, it uses the browser's MediaDevices API; the Monaco-style editor is used for coding-related questions; and the custom SVG is used to show analytics charts. No third-party JavaScript framework is integrated; hence, dependency exposure is reduced to zero installed packages, one can operate the prototype in any recent browser even without the build step, and the fear of framework version drift as a reproducibility concern is eliminated. Typography features Geist Sans (UI) and Instrument Serif (display headings); iconography uses the Lucide icon set; the colour system centres around Indigo-600 (#4F46E5) as the main accent colour with a nine-step scale and semantic colours designated for success, warning, and danger.

Application State and Routing
Everything about the application's state is stored in a single global object: AppState. This object consists of three main parts (student, proctor, admin) and a shared session context (current role, current screen, theme, density). Any change or mutation is made only by a special setState(patch) function, which does a deep merge of the patch and then initiates the render() cycle. Hence, the UI is always a true and consistent representation of the state. Routing is a pure function from (role, screen) pair to a rendered HTML string. The gate logic is centralised into a ROUTE_GUARDS table; for instance, the student exam page is accessible only when identityVerified is true, systemCheckPassed is true, and the exam window is open. Unsuccessful guards result in a user-friendly, contextual gate-blocked message rather than changing the location.

Core Services
The four main services are implemented as separate modules, each responsible for writing different parts of the USER record. The Identity Verification Service takes an ID document and a biometric photo, uses OCR to extract the name, performs face comparison to score the match, and updates the identity_verified field with the result. The Proctoring Engine takes a webcam frame every second during the exam period, analyses the six signals of Table II, and delivers alerts to the proctor dashboard via WebSocket with a 2 s delay. The Plagiarism Detection Engine is activated once a response is submitted; it divides the text into tokens, generates n-gram fingerprints, and matches them against the web corpus and the peer submission repository, recording per-response similarity scores along with the top source references for inline highlighting. The AI-Content Detection Service operates both on submission time and during essay composition time at intervals of 30s; it calculates perplexity and burstiness features as per Gehrmann, Strobelt & Rush (2019) and Tian & Cui (2023), runs a binary classifier giving P(AI | response), and refreshes the live Originality Indicator on the student panel. In a real-world scenario, the simulated face comparator would be substituted with a FaceNet model (Schroff, Kalenichenko & Philbin, 2015) deployed locally to prevent cross-border biometric data transfer, and the AI-content classifier with a fine-tuned RoBERTa model (Liu *et al.*, 2019).
By design, none of the four integrity engines exposes an editable prompt or scoring parameter to the examinee, which guards the platform against adversarial prompt editing during an active exam window. All detection logic — perplexity and burstiness feature extraction, classifier inference, and threshold comparison — executes server-side within the core-services layer; the presentation layer renders only read-only outcomes, such as the live Originality Indicator, and never transmits a prompt to the detector or receives model internals from it. Examinee input is therefore treated strictly as untrusted data rather than as instructions. Each captured response is time-stamped and written to an append-only field of the USER record at capture time, so any in-exam attempt to alter previously submitted text, inject control tokens, or re-prompt the classifier is rejected by the schema and logged as a proctor event. Combined with the Verity-locked

browser environment, which turns off developer tools, clipboard injection, and out-of-window navigation, this server-authoritative, immutable-input design ensures that the integrity scores attached to a session cannot be influenced by prompt manipulation on the examinee's device.

Evaluation Protocol

Verity was evaluated using two complementary instruments. Instrument 1 — System Usability Scale (SUS) is a validated 10-item Likert-scale instrument producing a composite score from 0–100 (Brooke, 1996); scores ≥ 70 indicate "Good" usability. Participants completed a structured task scenario for their role and a SUS questionnaire immediately afterward. Instrument 2 — Expert Heuristic Evaluation used three domain experts to independently score violations of Nielsen's ten usability heuristics (Nielsen, 1994) on a 0–4 severity scale, with inter-rater agreement reported as Krippendorff's α (acceptable threshold ≥ 0.60). Both instruments were administered on the functional Verity prototype. Ethical approval was obtained from the authors' institutional review board prior to data collection.

Participants and Task Scenarios

Through purposive sampling, 28 participants were selected from a single higher education institution: 18 students (Student role), 6 academic staff (Proctor role), and 4 educational technologists/administrators (Admin role). All participants except one reported having prior experience with online examinations. In addition, three domain experts were involved in the heuristic evaluation only (one HCI researcher, one EdTech developer, and one academic integrity officer). Participant demographics are summarised in Table III.

Table III: Participant Demographics

Role	N	Gender (M/F/Other)	Age Range	Prior Online Exam Experience
Student	18	10 / 8 / 0	18–28	18/18
Proctor	6	3 / 3 / 0	30–55	5/6
Admin	4	2 / 2 / 0	32–48	4/4
Total	**28**	15 / 13 / 0	18–55	27/28

Each participant completed a structured task scenario matched to their role, using a printed task card and a think-aloud protocol (Ericsson & Simon, 1993). The student scenario exercised all five student screens (S1–S5), including identity verification, system check, mixed-format examination, and results review. The proctor scenario exercised P1–P3, requiring participants to identify the highest-alert student, filter by High severity, review an incident timeline, and resolve a Medium-severity flag with a written note. In the admin scenario, A1–A3 were exercised; participants had to identify the lowest-integrity exam, locate all three highlight types in a submission, record a formal "Warning" ruling with justification, and identify outliers in the analytics scatter plot. Work sessions were 25 to 40 minutes long per participant.

RESULTS

SUS Study Results

SUS scores were calculated in the usual way, and their meaning was checked against the adjective rating scale of Bangor, Kortum & Miller (2008). Table IV summarizes the results.

Table IV: SUS Score Summary by Role

Role	n	Mean (SD)	Min	Max	Rating
Student	18	79.9 (5.4)	72.5	90.0	Good
Proctor	6	77.6 (4.9)	70.0	85.0	Good
Admin	4	84.7 (4.1)	80.0	90.0	Excellent
Overall	**28**	**80.1 (5.1)**	**70.0**	**90.0**	**Good**

The overall average SUS score was 80.1 (SD = 5.1), 10.1 points above the initial functional target of 70; therefore, the usability level was rated as "Good." Not a single score was below 70; additionally, 60.7% (17 of 28) of users rated the system "Excellent" or higher. The Admin role had the highest average score (84.7), which also shows how easy it was to read and review; while the Student role average (79.9) highlights the challenge of the five-screen gated flow; and the Proctor role average (77.6) reflects the dense nature of the interface in the live monitoring dashboard, which four proctor participants described as overwhelming at first sight before they became comfortable with it within 3 to 4 minutes of first use. A one-way ANOVA across the three roles yielded $F(2, 25) = 2.14$, $p = .139$, indicating no statistically significant difference in usability between roles and confirming that role-separated interfaces achieve comparable usability across fundamentally different task profiles.

It should be acknowledged that the participant pool, while adequate for a high-fidelity prototype usability test, is unevenly distributed across roles, comprising 18 students, 6 proctors, and 4 administrators. This skew mirrors the underlying population of an examination platform, in which examinees vastly outnumber the specialised staff who proctor sessions or administer an institution. The smaller proctor ($n = 6$) and admin ($n = 4$) subsamples necessarily limit the statistical power of between-role comparisons, and the ANOVA reported above should therefore be read as indicative rather than confirmatory. Nonetheless, both subsamples still meet the widely cited minimum of four to five participants for surfacing the majority of usability problems within a single, well-defined role (Nielsen, 1994), and they adequately cover the narrow, highly specialised proctor and administrator demographics. In effect, the student-weighted distribution concentrates measurement where the interface is most heavily exercised while retaining sufficient coverage of the staff-facing interfaces to expose their principal usability issues.

Expert Heuristic Evaluation

Three domain experts independently evaluated the prototype across all role interfaces using Nielsen's ten heuristics (Nielsen, 1994). After merging duplicate findings, 22 distinct violations remained. Inter-rater agreement on severity ratings was $\alpha = 0.71$, exceeding the acceptable threshold of 0.60 (Krippendorff, 2013). Severity distribution: 0 violations rated 4 (catastrophe), 2 rated 3 (major: H05 — no auto-save when leaving the essay question; H09 — identity capture accepted under poor lighting), and 20 rated 1–

2. Both Severity-3 issues were addressed before the SUS study: auto-save with a "Saved" toast was added to the essay flow, and a pre-capture luminance check ("Improve lighting") was added to the identity step. Of the remaining lower-severity issues, all were resolved except H20 (in-exam pause functionality), which was accepted as a limitation and documented as future work.

Functional Verification

Five scenario-based test cases were executed to confirm that core integrity thresholds are enforced correctly: (FC01) an identity match of 0.76 (below 0.80) correctly blocked the gate and prompted retry; (FC02) a tab-switch event was logged and propagated to the proctor feed within 2 s; (FC03) a Jaccard similarity of 0.42 against a peer submission produced a High-severity plagiarism alert with correct orange inline highlighting; (FC04) an AI probability of 0.91 produced a High-severity AI-content alert with correct blue inline highlighting; (FC05) submission of an essay with 0 words was correctly blocked with a warning modal. All five cases passed.

Results Summary

The evaluation establishes three findings. First, Verity achieves strong perceived usability across all three roles (overall SUS = 80.1, "Good"), satisfying the utility criterion U1. Second, role-separated interface design produces statistically comparable usability across fundamentally different task profiles ($F(2, 25) = 2.14$, $p = .139$). Third, no unresolved Severity-3 or Severity-4 heuristic violations remain in the final prototype, satisfying utility criterion U2.

DISCUSSION

Addressing the Identified Gap

The central claim of this work is that online assessment integrity is a multi-layer problem and that meaningful enforcement requires the layers to be unified at the system level. The evaluation provides evidence for this claim in three specific ways. First, the USER data model co-locates proctoring events, plagiarism scores, and AI-content probabilities in a single record, so the cross-dimensional reviews that current toolchains require to perform manually become single-query operations. All six proctor participants completed an end-to-end flag review workflow within a single interface — a task that today requires switching between a proctoring platform and a separate case management system. Second, the simultaneous three-type inline highlighting in the admin report has no direct equivalent in current commercial offerings, where plagiarism and AI-content findings are surfaced in separate reports and require reviewers to integrate evidence mentally; all four admin participants located all three highlight types within a single reading. Third, the live Originality Indicator, presented during essay composition, introduces a deterrence mechanism absent from existing tools: it surfaces integrity findings only after submission. Although the current study does not isolate the indicator's deterrent effect, three student participants spontaneously commented during the think-aloud protocol that the indicator made them more conscious of their writing process; this suggests an isolated deterrence study.

Comparison with Existing Tools

Table V shows how Verity stacks up against ProctorU, Turnitin, and Originality.ai in terms of integrity and interface integration capabilities.

Table V: Capability Comparison

Capability	ProctorU	Turnitin	Originality.ai	Verity
AI-based proctoring	Yes	—	—	Yes
Plagiarism detection	—	Yes	—	Yes
LLM-content detection	—	Partial	Yes	Yes
Unified integrity record	—	—	—	Yes
Role-separated UI	Partial	Partial	—	Yes
Live originality indicator	—	—	—	Yes
Three-type inline highlighting	—	—	—	Yes
Adjudication workflow	Partial	—	—	Yes
Live human proctoring	Yes	—	—	Future
LMS integration (LTI)	Yes	Yes	Partial	Future

Verity is the only platform among those compared that can simultaneously address all three integrity dimensions and make them accessible via integrated, role-specific user interfaces. The comparison does not, however, imply that the products are equal in production. For example, ProctorU provides a live human proctoring element and very mature LMS integration, which are explicitly planned as future work for Verity.

Privacy, Ethics, and Limitations

Any system that records webcam footage, captures biometric data, and monitors keystroke patterns walks a fine line between educational utility and privacy (Flores Zavaleta, 2024; Han *et al.*, 2024). Verity's role permission matrix sets tight limits (proctors cannot see response content; students cannot see aggregate integrity data); webcam frames are used only to identify events and are not stored as raw video. A production release, however, would require explicit informed consent for biometric processing, a Data Protection Impact Assessment, and a bias-audited face-detection model, since face-comparison systems have been shown to exhibit varying levels of accuracy across demographic groups (Buolamwini & Gebru, 2018). AI-content classifiers can also generate a significant number of false positives for non-native English writers (Liang *et al.*, 2023). They should therefore be considered one of several signals rather than definitive evidence of wrongdoing. The main shortcomings of the present investigation are: (i) Verity is a high-fidelity prototype with simulated backend services, so the user experience under real network latency and submission content may not be the same; (ii) the 28-participant sample consists of students only from one institution and restricts generalisability; (iii) the measurement of usability was a single 25

to 40-minute session and does not take into consideration longitudinal effects; and (iv) the research assesses usability and functional accuracy only, not the efficacy in reducing real misconduct, which requires a controlled deployment study.

Conclusion and Future Work

This paper presented Verity, an integrated multi-role online examination platform that unifies AI-based proctoring, plagiarism detection, and LLM-based content analysis within a single cohesive architecture. The work was motivated by the architectural fragmentation of existing tools, which forces institutions into multi-vendor toolchains with blind spots at the boundaries between systems. Verity was developed using the Design Science Research methodology and evaluated through a 28-participant SUS study and a three-expert heuristic evaluation. The system achieved an overall SUS mean of 80.1 (Good) with no participant scoring below 70 (utility criterion U1 satisfied), produced statistically comparable usability across all three roles ($F(2, 25) = 2.14$, $p = .139$), and resolved all Severity-3 heuristic violations prior to evaluation (utility criterion U2 satisfied).

Five directions are identified for future work. First, production backend integration replacing the simulated services — in particular, a bias-audited face recognition model (Schroff, Kalenichenko & Philbin, 2015) and a fine-tuned RoBERTa-based AI-content classifier (Liu *et al.*, 2019). Second, a controlled multi-institution deployment study measuring proctor alert precision/recall, classifier accuracy against ground-truth labels, longitudinal usability, and the deterrence effect of the live Originality Indicator. Third, LMS integration through the latest IMS Global Learning Tools Interoperability (LTI 1.3) standard, ensuring seamless single sign-on and grade passback. Fourth, mobile-native adaptation enables delivery on smartphones and tablets, with additional integrity signals that can be fetched from various sensors on the device. Fifth, expansion of the integrity tuple to comprise keystroke-dynamics authorship verification and adaptive per-student question generation. Together, these steps establish a reliable research agenda grounded in the already validated architectural foundation of Verity.

REFERENCES

Balalle, H. & Pannilage, S. (2025). Reassessing academic integrity in the age of AI: A systematic literature review on AI and academic integrity. *Social Sciences & Humanities Open*, Elsevier.

Bangor, A., Kortum, P. T. & Miller, J. T. (2008). An empirical evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction, 24*:574–594.

Brooke, J. (1996). SUS — a quick and dirty usability scale. In: *Usability Evaluation in Industry*, edited by Jordan, P. W., Thomas, B., Weerdmeester, B. A. & McClelland, I. L. London: Taylor & Francis, 189–194.

Buolamwini, J. & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. *Proceedings of the Conference on Fairness, Accountability, and Transparency* 2018:77–91.

Ericsson, K. A. & Simon, H. A. (1993). *Protocol Analysis: Verbal Reports as Data*, rev. ed. Cambridge, MA: MIT Press.

Evangelista, E. D. L. (2025). Ensuring academic integrity in the age of ChatGPT: Rethinking exam design, assessment strategies, and ethical AI policies in higher education. *Contemporary Educational Technology*.

Flores Zavaleta, C. E. (2024). A systematic review of literature on e-proctoring technologies for examination supervision in higher education. *Perfiles Educativos*.

Gehrmann, S., Strobelt, H. & Rush, A. M. (2019). GLTR: Statistical detection and visualisation of generated text. *Proceedings of the ACL System Demonstrations* 2019:111–116.

Han, S., Nikou, S. & Ayele, W. Y. (2024). Digital proctoring in higher education: a systematic literature review. *International Journal of Educational Management*, Emerald.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*:75–105.

Krippendorff, K. (2013). *Content Analysis: An Introduction to Its Methodology*, 3rd ed. Thousand Oaks, CA: SAGE.

Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J. (2023). GPT detectors are biased against non-native English writers. *Patterns, 4*:100779.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Malhotra, M. & Chhabra, I. (2026). Ensuring academic integrity through automated online exam proctoring: a decade-long systematic review. *Discover Education*, Springer.

Nielsen, J. (1994). Heuristic evaluation. In: *Usability Inspection Methods*, edited by Nielsen, J. & Mack, R. L. New York: Wiley, 25–62.

Ogwueleka, F. N. (2025). Plagiarism detection in the age of artificial intelligence: current technologies and future directions. In: *AI and Ethics, Academic Integrity and the Future of Education.*

Peffers, K., Tuunanen, T., Rothenberger, M. A. & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*:45–77.

Saltzer, J. H. & Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE, 63*:1278–1308.

Schroff, F., Kalenichenko, D. & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2015:815–823.

Sozon, M., Alkharabsheh, O. H. M., Fong, P. W. & Chian, S. B. (2024). Academic integrity violations in higher education: a systematic literature review from 2013–2023. *Journal of Applied Research in Higher Education*, Emerald.

Sozon, M., Alkharabsheh, O. H. M., Fong, P. W. & Chian, S. B. (2026). Exploring three decades of key themes and trends in academic misconduct in higher education institutions. *Journal of Applied Research in Higher Education*, Emerald.

Tian, E. & Cui, S. (2023). GPTZero: Towards detection of AI-generated text using zero-shot and supervised approaches. *arXiv preprint arXiv:2310.05169*.

Topuz, A. C. (2024). Students' acceptance of and preferences regarding online exams: A systematic literature review. *Educational Technology Research and Development*.