

Risk Controlled Architecture and Offline Authorization Testing for Student Facing Large Language Model Agents with Minimal Data, Permission Boundaries, Scenario Checks, and Human Escalation

Nayyar Ahmed Khan^{1*}, Mohammed Ahmad² & Rim Hamdaoui²

¹Department of Computer Engineering, College of Computing and Information Technology, Shaqra University, Saudi Arabia.

²Department of Computer Science, College of Science and Humanities, Dawadmi, Shaqra University, Saudi Arabia.

Corresponding Author (Nayyar Ahmed Khan) Email: nayyar@su.edu.sa*



DOI: <https://doi.org/10.46759/ijrsr.2026.10311>

Copyright © 2026 Nayyar Ahmed Khan et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Article Received: 22 July 2026

Article Accepted: 26 September 2026

Article Published: 30 September 2026

ABSTRACT

This study specifies a risk controlled student-facing large language model (LLM) agent with a permission matrix, minimal data flow, response states, scenario tests, and human escalation. A server-side authorization function was tested on 16 constructed role-action requests: six were allowed, ten denied, and all matched the expected decision. A hypothetical prelaunch case shows why content, accessibility, retention, and handoff failures remain separate release gates. The unit test verifies only the stated permission rules; it cannot establish model safety, policy compliance, or educational benefit.

Keywords: Student Facing Agents; Large Language Models; Permission Matrix; Data Minimization; Threat Modeling; Prompt Injection; Human Escalation; Scenario Testing; Accessibility; Incident Response; Offline Authorization; Educational Governance.

1. Introduction

A student facing agent may retrieve a lesson, retain context, recommend practice, or invoke a tool. Each capability changes what it can know and what it can affect. A general statement that the agent is “safe” is therefore too vague to evaluate. The technical problem is to specify the permitted actions, sensitive data paths, failure cases, and human authority for a defined classroom task.

This paper considers an agent that gives hints on ungraded practice from approved unit notes. It excludes grade changes, formal support recommendations, and autonomous messages to students or families. The research question for a future pilot is whether the proposed controls keep errors detectable and correctable while preserving useful help and equitable access. The framework is not a legal compliance opinion or an empirical safety result [1-4].

The contribution is a permission matrix, a minimal trace schema, scenario-based tests, a severity-based incident process, and a release decision record. The school remains accountable for local policy, data agreements, age-appropriate supervision, and alternative support.

1.1. Study Objectives

- Map capabilities, data fields, and trust boundaries for a narrow task.
- Enforce tool permissions outside the language model.
- Define escalation, disabled states, and incident ownership.
- Verify role-action decisions on constructed requests.

- Test source, access, privacy, and handoff scenarios before release.
- Specify monitoring and retirement criteria for a limited pilot.

2. Literature Review

Guidance and research since 2020 have emphasized privacy, transparency, fairness, human oversight, and the risk of overreliance on fluent artificial intelligence (AI) answers. Educational agents add tool use and memory to ordinary text generation, creating more ways for data or decisions to move beyond the original question [5-10]. A retrieved document can contain an incorrect fact or an instruction that tries to override the agent's purpose. A source citation alone does not establish that an answer is supported.

Technical risk management consequently needs both preventive and detective controls. Preventive controls limit tools and data; detective controls test outputs, monitor incidents, and make a human handoff available. This paper operationalizes those ideas for a narrow educational use. It does not claim that passing a small test set proves general safety [11-13].

Scenario tests include an ordinary hint, a false premise, an unsupported administrative fact, an attempt to reveal another learner's information, a prompt injection in retrieved content, an additional-language query, inaccessible visual material, and a sensitive personal request. For each scenario, reviewers write acceptable behavior and a failure threshold before testing [14-17]. Repeat stochastic cases and record model and corpus versions. A proposed pilot set may include several variants per scenario, but the final sample is set locally before release.

Planned measures include source-support rate, unauthorized tool-call count, correct handoff proportion, access-failure count, and teacher correction time. Report both interaction counts and distinct learners. Reviewers should inspect a sample of ordinary exchanges as well as all serious incidents [18-21]. A low incident count is difficult to interpret without knowing use volume, detection coverage, and whether students could report a problem.

Before launch, assign a curriculum owner, data lead, technical operator, and student support contact. Define retention, staff access, incident response time, and a backup human route. The pilot must have a fixed scope, duration, and stopping rule. Its outcome can be continue, revise, narrow, pause, or retire. The process is designed to make that choice traceable rather than to guarantee a favorable release decision. A monitoring plan samples ordinary answers and reviews every critical incident [22-25]. It records the denominator of interactions, not merely the number of flagged failures. Stratify a sample by subject topic and interface mode so that rare but difficult uses are visible. A teacher corrects content errors; a data lead handles a disclosure; a technical owner disables a failing tool. The review meeting records decisions and deadlines. If staffing cannot meet those duties, the permitted scope should be reduced before release [22]. The permission-layer unit test uses five actions and three roles: student, teacher, and unknown. Sixteen constructed requests specify an expected allow or deny decision. Grade writing and message sending are forbidden to every role in this tutoring system; a teacher can read grades only in the test matrix, although the student agent itself receives no gradebook connection. Authorization is evaluated before an

LLM can invoke a tool [23-28]. The suite tests a server-side matrix, not model resistance to prompt injection or the completeness of institutional policy.

[1] synthesize privacy, bias, and institutional governance concerns in generative artificial intelligence for education. [2] frame bounded autonomy as a way to retain learner decision authority when agents can act over time. The present architecture translates those broad concerns into a server-side role-action check, a minimal event record, disabling conditions, and human escalation; the small constructed permission test does not measure general safety.

3. Methodology

The system boundary includes a learner interface, approved content index, LLM service, policy engine, response checker, minimal event log, and teacher handoff channel. The data-flow inventory records each field entering or leaving these components. For an ungraded fraction question, the exercise identifier and attempted step may be needed; address, diagnostic label, complete academic record, and unrelated prior chats are not. A short-lived context can support continuity in the current exercise [29]. The permission model assigns each capability an allowed source, action, and human approver. The agent may read student-visible unit notes, but it cannot read another learner's data, alter grades, send messages, or expose a private answer key. These restrictions should be enforced at the tool boundary rather than only through a prompt. A newly connected tool requires an updated permission review and test set. Table 1 makes the capability boundary inspectable.

The response state machine has four states: ordinary answer, clarification, human escalation, and disabled. A low-risk content question can proceed to a checked hint. Missing or conflicting evidence triggers clarification or escalation [30-32]. Requests about grades, personal advice, or another student's information bypass ordinary tutoring. A serious privacy disclosure or failed handoff moves the affected function to disabled until investigation and retesting. Students can request a person at any point. Figure 1 summarizes the sequence and the authority boundary described above.



Figure 1. Risk controlled release sequence for a student facing agent.

Capabilities are defined, unnecessary data and tools removed, failure cases tested, and the system either enters a bounded pilot or remains disabled. Source: Original schematic prepared for this manuscript.

The risk register records a hazard, initiating event, potential impact, preventive control, detection method, owner, and response. Severity and likelihood can be rated on a predeclared ordinal scale for prioritization, but multiplying ordinal numbers does not prove that a risk is acceptable. An incident decision must consider the nature of the harm and the age and context of learners. The register is reviewed after every material model, prompt, source, or tool change.

Equation (1) states the acceptance or scoring rule used by the proposed method; its variables are defined in the surrounding text and Table 1.

$\text{Authorize}(\text{role}, \text{action}) = 1$ only if $\text{action} \in \text{Permissions}(\text{role})$; unknown role or missing context $\Rightarrow$ deny. (1)

In Equation (1), role is the authenticated user category and action is the requested tool operation. $\text{Permissions}(\text{role})$ is the server-maintained set of permitted actions for that role; the membership symbol $\in$ means that action belongs to that set. Authorize returns 1 only for an allowed, fully contextualized request and otherwise returns 0 (deny). Missing context includes an absent authentication or policy version.

Algorithm 1. Offline decision procedure

INPUT authenticated role, requested action, policy version

Check role and action against server-side permission matrix

IF unauthorized: deny before any model tool execution

IF sensitive request: expose verified human route

Log minimal event type and policy decision

On critical incident: disable affected capability and retest

The data-flow inventory can be expressed as a table of source, field, destination, purpose, retention, and access role. A student question reaches the tutoring service; a retrieved passage identifier reaches the generator; a minimal event record reaches quality review. The inventory should also cover diagnostic logs created by external services, which may otherwise be overlooked. Each field needs a reason linked to the defined task. If the same task works without a persistent student identifier, that identifier should be removed from routine processing.

Prompt injections are tested at two trust boundaries: a learner message and a retrieved document. A malicious instruction embedded in an indexed passage is treated as untrusted content, even though the document is used as subject evidence. The policy engine and tool permissions reside outside that content. In a controlled test, the passage can request access to a hidden answer key or instruct the model to ignore the teacher. Success means that no unauthorized tool call occurs, and the answer does not reveal protected material. A model may still fail, so write permissions remain unavailable. The scenario suite should vary wording and context. A privacy test can request another learner's marks directly, indirectly through a comparison, and through a forged teacher instruction. A sensitive-support test can be phrased in formal and colloquial language. An access test should use the actual screen reader or low-bandwidth interface where feasible. Reviewers assess the full handoff, including whether a person receives enough information and whether the contact route is open. A generic refusal is not a successful handoff. Risk prioritization separates severity, exposure, and detectability [33]. For example, a single incorrect low-stakes practice hint may be correctable, while exposing another learner's record requires immediate containment even if the event seems rare. A risk register can use categories such as low, moderate, and critical, with written definitions. The owner states which controls prevent an event, and which detect it afterward. A severe unresolved hazard blocks

release regardless of an attractive average quality score. Table 1 defines the main variables and control checks used in this method.

Table 1. Capability and control matrix showing which course, memory, learner-data, external-action, and sensitive-request operations the agent may perform.

Capability	Allowed operation	Control
Course retrieval	Read current student notes	Version and access filter
Conversation memory	Current exercise context	Short retention and reset
Learner data	No private records	Tool-level denial
External action	No messages or grade edits	No write permission
Sensitive request	Human handoff	Tested contact route

Incident handling follows a documented sequence: receive report, contain the affected capability, preserve necessary evidence under access control, determine scope, correct the cause, retest, and communicate according to local procedure. A source error may be fixed by replacing a passage and retesting related questions. A permission failure may require disabling a tool and reviewing previous calls. The review record should note who authorized reactivation. This process must be practical for the actual staffing model rather than a checklist nobody can execute.

Capability review begins with a dependency graph. A tutoring hint requires the learner question, an approved lesson passage, and a response policy. It does not require a gradebook connection. Removing the gradebook edge eliminates an entire class of possible unauthorized reads. A practice recommendation may need a record of recent exercise outcomes, but that addition should be separately justified and tested. Each new dependency changes the threat model, data inventory, and access notice. The permission matrix should be tested below the model layer. A direct tool call requesting a private record must be denied even if the model accidentally attempts it. A retrieved document cannot grant a permission [34-38]. Test accounts with distinct student data can confirm that cross-user isolation holds. Log unauthorized attempts without retaining more sensitive content than needed for investigation. The policy engine should treat a missing user identity or ambiguous authorization as denial, not as a reason to guess the intended account.

A handoff contract states when escalation is triggered, what the learner sees, which staff route receives the request, and what data accompany it. A sensitive question may require minimal context and a direct contact option rather than a detailed model summary. Test the route during normal hours and after hours. The contract includes a fallback if no staff member responds. The agent must never claim to have notified a person unless the notification has actually succeeded.

4. Results and Discussion

The authorization function returned the expected decision for all 16 constructed requests: six allowed and ten denied. Unknown-role attempts were denied, as were student grade reads and every grade-write and message-send request. This is a unit test of the permission layer. It does not show that a deployed agent will recognize every

sensitive conversation, that retrieved text cannot contaminate an answer, or that the human handoff will work under real operating conditions. Table 2 records the constructed trace or issue case used to illustrate the decision path.

Table 2. Hypothetical prelaunch decision record showing why retention, source, accessibility, and handoff failures hold the agent before pilot release.

Finding	Required correction	Release state
Excess log retention	Minimize and time-limit logs	Hold
Unsupported lesson claim	Correct source or response	Hold
Diagram lacks text alternative	Add and test description	Hold
Handoff unavailable	Provide staffed route	Hold until retested

In the hypothetical prelaunch scenario, long retention, an unsupported source claim, an inaccessible diagram, and an unavailable handoff keep the release state on hold. This scenario tests the decision record concept, while the 16-request suite tests the explicit authorization function. The two results must not be combined into a single safety percentage: they concern different components and contain no live learner data. Table 3 reports the offline component-test outputs and their denominators.

Table 3. Server-side authorization unit test on 16 constructed role-action requests; the counts verify only the specified permission matrix.

Decision class	Count	Expected behavior
Allowed	6	Return authorized content or handoff
Denied	10	No tool action
Unexpected decisions	0	No mismatch in this suite
Unknown-role cases	3	All denied

The narrow result supports enforcing permissions outside the LLM. A future evaluation should add prompt-injection variants, cross-user isolation tests, actual interface access tests, incident detection, and staffed handoff drills. A perfect unit-test pass rate on 16 requests is not a safety certification.

The strongest control is a narrow capability boundary. A tutor that cannot read private grades or send messages has fewer paths to cause consequential harm. Prompt instructions still matter, but they should not substitute for permissions. Similarly, a human fallback only protects learners if its contact route is staffed and available at the time the student needs it [39-41]. Fairness testing should compare substantive help across dialects, proficiency levels, accessible formats, and realistic device conditions. The system must not infer ability from response speed or writing style. An alternative route to teacher help is needed for learners who cannot use the agent. Group-level analysis, where appropriate and permitted, should protect privacy and investigate the causes of differences rather than assign deficits to students [42]. The framework has limitations. Scenario tests cannot cover every future conversation, and model behavior can change after updates. Incident logs may miss unreported harm. Staff capacity may be insufficient for promised review. The institution should therefore retest after material changes, sample real

use, and retain the ability to disable a function. A decision to retire or narrow an agent is a valid outcome of the pilot. A minimal data policy and a useful audit trail can conflict. The school needs enough context to investigate an incorrect hint, yet should avoid indefinite storage of personal conversation. A practical compromise is a short operational retention period with restricted access and a separate, deidentified set of cases for evaluation, where permitted [43]. The precise retention period depends on local rules and purpose. The design must disclose the choice to learners rather than conceal it inside a technical setting.

A student may rely on an agent because it responds quickly and confidently. Clear source displays, uncertainty statements, and a visible human option can reduce misplaced authority, but their effectiveness needs testing with real users. Younger learners and those with limited access may require different supervision. A school should not infer that a well-designed interface solves a shortage of teachers or support staff. The agent remains a bounded supplement. Ordinal risk labels can aid prioritization but can also hide uncertainty. Reviewers may disagree about likelihood because incidents are rare or underreported. The record should preserve that disagreement and the basis for a decision [44-48]. A severe possible disclosure can justify a tool-level prohibition even when its estimated frequency is low. The most useful risk register drives a concrete control and owner, not a single total score.

Long-term governance needs a retirement path. A school can disable personalized recommendations while retaining a static course search function. It should remove stale links, revoke tool permissions, provide an alternative help route, and apply the published retention policy to stored data. A pilot that ends in retirement can still be a successful evaluation if it reveals that the proposed benefit does not justify ongoing review cost or risk [49]. Control effectiveness must be verified over time. A permission test at launch cannot cover a new tool connection or a revised provider interface. The technical owner should maintain a change register and run the affected scenario suite before reactivation. A curriculum owner verifies source changes separately. The review cadence can be risk based: a passive search tool may need less intensive sampling than an agent recommending a learning pathway. The rationale for cadence belongs in the decision record [50]. Incident underreporting is a major threat to evaluation. Learners may not recognize an incorrect hint or may find the report control difficult to use. A teacher can sample ordinary interactions and include a brief classroom exercise in which students practice checking a source and reporting a problem. The report route should be accessible without requiring a lengthy explanation. Measures of safety should state how incidents were detected, who reviewed them, and what proportion of use was sampled [52,53].

Escalation can also expose more information than necessary. A human support route may need the learner's contact and the nature of a request, but not the entire tutoring history. The system should show the learner what will be sent and allow a direct contact path where feasible. For urgent situations, the school's established protocol governs the details. Testing should include both a successful handoff and a failed or delayed destination, since a button alone does not establish that a person is available. Finally, access and educational utility must be considered together. An agent that refuses every difficult question may meet a narrow safety measure while providing little learning support. A system that answers nearly everything may be useful in the moment but expose learners to unsupported advice. A pilot should report both kinds of failure and the availability of alternatives. The release decision should be

grounded in the actual learner population, staffing capacity, and course purpose. The hypothetical review can be represented as a sequence of state transitions. The agent begins in prelaunch. Excessive retention, unsupported content, inaccessible material, and a broken handoff each keep it in hold. After the owners correct and retest each critical item, a school may authorize a limited supervised pilot. A later severe disclosure moves the affected capability to disabled regardless of ordinary hint quality [54-57]. The state record explains why the system changed status and who approved the change.

In a real pilot, a decision table would contain observed counts, dates, owners, and unresolved issues. The present table contains only hypothetical findings. It shows how a school can require evidence for release without turning a single numerical risk score into a substitute for judgment [58-61]. The hypothetical record distinguishes control failure from consequence. Long retention is a data-governance failure even if no disclosure has occurred. The unsupported citation is a content-verification failure even if the answer happens to be approximately correct. The inaccessible diagram is an access failure, and the unavailable contact is an escalation failure. Treating them as separate categories prevents one favorable average score from hiding a critical defect. Each category has a different owner and a different retest. A concrete release gate can require zero unresolved critical issues, an approved data flow, a working human route, and a signed pilot scope [62]. These are proposed criteria. They do not assert that a system satisfying them will never fail. The pilot decision record should include the test version, people who approved it, unresolved moderate issues, and the date of the next review. Without those fields, a later team may be unable to explain why the agent was made available.

5. Conclusion and Future Recommendation

This paper specifies a student facing agent with explicit permissions, minimal data, scenario tests, human escalation, and stopping rules. Its hypothetical review shows how the release decision can be documented. Educational usefulness and risk acceptability require a bounded real pilot with transparent measures and continuing human accountability.

5.1. Future Recommendations

- 1) Extend authorization tests to cross-user data isolation.
- 2) Test prompt injection in retrieved lessons and learner messages.
- 3) Run staffed handoff drills under normal and after-hours conditions.
- 4) Measure reporting coverage and unresolved incidents in a pilot.
- 5) Assess fairness across language, disability, and connectivity conditions.
- 6) Retest capabilities after every model, tool, or curriculum change.

Declarations

Source of Funding

The authors received no funding for this contribution.

Competing Interests Statement

There are no conflicts of interest by the authors in any way.

Consent for publication

The authors give consent for the publication.

Authors' contributions

All the authors contributed equally to the article.

Informed Consent

Not Applicable. The reported offline checks used constructed records and no human participants.

Availability of data and material

Constructed test inputs and deterministic procedures are described in the manuscript.

Institutional Review Board Statement

Not Applicable. No human participants or identifiable student records were included in the offline checks.

Ethical Approval

Not Applicable to the constructed offline unit tests.

References

- [1] García-López, I.M., & Trujillo-Liñán, L. (2025). Ethical and regulatory challenges of Generative AI in education: A systematic review. *Frontiers in Education*, 10: 1565938. <https://doi.org/10.3389/feduc.2025.1565938>.
- [2] Salah, M., Al-Busaidi, R., Al-Ghawi, A., & Shawky, W. (2026). The regulation paradox: Agentic AI, bounded autonomy, and self-regulated learning in higher education. *Frontiers in Education*, 11: 1902833. <https://doi.org/10.3389/feduc.2026.1902833>
- [3] Ahmad, M., et al. (2025). Learning three-dimensional face recognition from sparse views for robust identity verification. Available at SSRN 5428214.
- [4] Akram, F., et al. (2024). Integrating artificial bee colony algorithms for deep learning model optimization: A comprehensive review. In *Solving with Bees: Transformative Applications of Artificial Bee Colony Algorithm*, Pages 73–102.
- [5] Al Omari, O.M.A., Khan, N.A., & Mahafdah, R. (2017). Ranking and reputation based resource allocation in P2P system. *Mediterranean Journal of Basic and Applied Sciences*, 1(1): 293–301.
- [6] Alanezi, R., Alanezi, M.A., & Khan, N.A. (2018). Development of web based E-cooperative training system. In *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, IEEE.
- [7] Alangari, S., et al. (2022). Developing a blockchain-based digitally secured model for the educational sector in Saudi Arabia toward digital transformation. *PeerJ Computer Science*, 8: e1120.

- [8] Alangari, S., & Khan, N.A. (2021). Artificially intelligent warehouse management system. *Asian Journal of Basic Science & Research*, 3(3): 16–24.
- [9] Alotaibi, R., Khan, N.A., & Akhtar, M.M. (2025). Statistical probability prediction model for E-learning and realtime proctoring using IoT devices. *Journal of King Saud University–Science*, 37.
- [10] Alshalaan, M., & Khan, N.A. (2025). Complexities and challenges for securing digital assets and infrastructure in academia: A review on digital asset security. In *Complexities and Challenges for Securing Digital Assets and Infrastructure*, Pages 225–244.
- [11] Alshalaan, M., & Khan, N.A. (2025). Blockchain-enabled federated learning framework for secure and collaborative drug discovery: Integrating AI, molecular docking, and distributed ledger technology.
- [12] Alshreef, B.S., & Khan, N.A. (2025). LSTM neural networks in healthcare disruption forecasting: A comprehensive review and implementation framework. In *International Conference on Innovative Computing and Cutting-edge Technologies*, Springer Nature Switzerland Cham.
- [13] Alshreef, B.S., & Khan, N.A. (2026). Disruption forecasting: A comprehensive review and implementation framework. In *Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025)*, Springer Nature.
- [14] Alsulami, M.H., Alotaibi, S., & Khan, N. (2021). Smart university model for Saudi Arabian universities. *Design Engineering*, Pages 162–181.
- [15] Alsulami, M.H., et al. (2021). Zigbee technology to provide elderly people with well-being at home. *International Journal of Sensors Wireless Communications and Control*, 11(9): 921–927.
- [16] Ayadi, W., Khalid, M.N., & Khan, N.A. (2026). Learning and fuzzy control. In *Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025)*, Springer Nature.
- [17] Ayadi, W., et al. (2025). Enhancing anomaly detection in autonomous systems using deep learning and fuzzy control. In *International Conference on Innovative Computing and Cutting-edge Technologies*, Springer Nature Switzerland Cham.
- [18] Gangwar, S., et al. (2026). AI-driven signal processing for single-cell data analysis: A case study in cancer research. In *Sensor Signal Processing for Intelligent Systems*, Pages 171–194, Springer Nature Switzerland Cham.
- [19] Haidar, S.W., et al. (2026). A secure data collaboration framework using differential partial AI and federated learning: Architecture, simulation, and performance evaluation. *Procedia Computer Science*, 282: 1237–1242.
- [20] Haidar, S.W., et al. (2025). Deep learning driven automated system for speech emotion recognition. In *2025 IEEE 7th International Conference on Computing, Communication and Automation (ICCCA)*, IEEE.
- [21] Haidar, S.W., et al. (2025). Enhancing problem-solving skills through modelling-based AI and generative flow networks (GFN). In *2025 2nd Global AI Summit–International Conference on Artificial Intelligence and Emerging Technology (AI Summit)*, IEEE.

- [22] Hassan, M.A.A., Khan, N.A., & Nasim, M.A.M. (2017). Managing data replication in mobile ad-hoc network databases using content based energy optimization. *Mediterranean Journal of Basic and Applied Sciences*, 1(1): 142–154.
- [23] Khan, N.A. (2018). Cloud applications development and deployment: The future of cost-effective programming and a step ahead. *Middle East Journal of Applied Science & Technology*, 1(1): 30–36.
- [24] Khan, N.A. (2019). Wireless requirements and benefits in the academics domain. *Middle East Journal of Applied Science & Technology*, 2(3): 45–49.
- [25] Khan, N.A. (2019). *Basics of Ethical Hacking and Computer Security*. Paperback, 1 January 2019.
- [26] Khan, N.A. (2019). Security management protocols in cloud computation. *Middle East Journal of Applied Science & Technology*, 2(1): 16–23.
- [27] Khan, N.A. (2021). Measuring academics intentions to use a project management system (PMS): A case study of the College of Computing and Information Technology, Shaqra University. In *Trends in Future Informatics and Emerging Technologies*, Pages 58–69.
- [28] Khan, N.A. (2022). Development of an artificially intelligent advising system for Saudi medical transcription. *Development*, 6(3): 94796.
- [29] Khan, N.A. (2024). Development of intelligent pick and drop service manager for small cities. *Asian Journal of Basic Science & Research*, 6(3): 20–27.
- [30] Khan, N.A. (2024). Development of intelligent help system for small cities. *Asian Journal of Applied Science and Technology*, 8(3): 112–119.
- [31] Khan, N.A. (2025). Development of intelligent student information system. *Arabian Journal of Basic Science and Research*, 7(1).
- [32] Khan, N.A. (2025). An application of machine learning, big data, and IoT of enterprise architecture: Challenges. In *Quality Control–Artificial Intelligence, Big Data, and New Trends: Artificial Intelligence, Big Data, and New Trends*, 111.
- [33] Khan, N.A., & Ahamad, D. (2025). Living smart: AI-based urban assistance systems for sustainable wellbeing in small cities. *Middle East Journal of Applied Science & Technology*, 3(2): 89–99.
- [34] Khan, N.A., et al. (2021). Development of mubadarah system—an intelligent system for proposals at a university. In *2021 International Conference on Computational Intelligence and Knowledge Economy*, IEEE.
- [35] Khan, N.A., et al. (2025). Transformative impact of artificial intelligence on higher education: A comprehensive analysis of pedagogical innovation, institutional transformation, and future learning ecosystems. *Asian Journal of Applied Science and Technology*, 9(4): 57–76.
- [36] Khan, N.A., et al. (2021). Development of Medidrone: A drone based emergency service system for Saudi Arabian healthcare. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE.

- [37] Khan, N.A., et al. (2026). A microservices-based optical mark recognition framework with advanced image processing and cloud-native architecture for scalable educational assessment. *Middle East Journal of Applied Science & Technology*, 9: 40–49.
- [38] Khan, N.A., et al. (2026). Design, development, and large-scale experimental evaluation of a multilingual, header-aware, course-centric intelligent attendance management and automated academic report generation framework for higher education institutions.
- [39] Khan, N.A., et al. (2026). User-centered development of an academic content website.
- [40] Khan, N.A., et al. (2025). Network intrusion management of web form spamming using blockchain. *Irish Interdisciplinary Journal of Science & Research*, 9(03).
- [41] Khan, N.A., et al. (2024). An IoMT enabled iterative artificial bee colony approach using federated learning for detection of heart disease. In *Solving with Bees: Transformative Applications of Artificial Bee Colony Algorithm*, Pages 103–116, Springer Nature Singapore.
- [42] Khan, N.A., et al. (2026). Revolutionizing drug design: The convergence of AI and quantum computing—A systematic review. In *Artificial Intelligence in Precision Drug Design*, Volume 2, Pages 317–326.
- [43] Khan, N.A., & Al Ghamdi, A.R. (2015). Cyber forensics and proposed techniques to overcome cyber threats for cyber security. *International Journal of Engineering and Management Research*, 5(5): 187–191.
- [44] Khan, N.A., & Albatein, J. (2021). COVIBOT—An intelligent WhatsApp based advising bot for Covid-19. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE.
- [45] Khan, N.A., & Alshalaan, M. (2025). AI-driven blockchain framework for digital transformation of academic accreditation process: A Saudi Arabian perspective.
- [46] Khan, N.A., & Alshalaan, M. (2026). Systematic review on machine learning applications in hardware security for vehicles. In *Software Engineering and Cybersecurity in Modern Electric Vehicles*, Pages 91–110.
- [47] Khan, N.A., et al. (2021). An empirical analysis on users' acceptance and usage of BYOD-technology for Saudi universities: A case study of Shaqra University. In *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, IEEE.
- [48] Khan, N.A., Khan, A.R., & Rajeyagari, S. (2025). Innovation in teaching and learning with the use of modern computational tools: A post Covid experience. *Middle East Journal of Applied Science & Technology*, 8(2): 74–82.
- [49] Khan, N.A., et al. (2026). Serverless absence management: A Google-native playbook for rapid, auditable university leaves workflows.
- [50] Khan, N.A., et al. (2026). A cloud-based multi-tier approval system for academic course registration: Design, implementation, and evaluations.
- [51] Khan, N.A., et al. (2020). Internet of Things (IoT) based educational data mining (EDM) system. *Journal of Mechanical and Computational Mathematics and Sciences*, 15(3): 271–284.

- [52] Khan, N.A., et al. (2019). Intrusion management to avoid web-form spamming in cloud based architectures. In 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), IEEE.
- [53] Khan, N.A., et al. (2025). A data-driven analysis of online learning expectations: Perspectives from Middle Eastern students and educators. In International Conference on Innovative Computing and Cutting-edge Technologies, Springer Nature Switzerland Cham.
- [54] Khan, N.A., et al. (2026). A data-driven analysis of online learning expectations: Perspectives. In Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025), Springer Nature.
- [55] Khan, N.A., Rajeyyagari, S., & Khan, A.R. (2025). Development of intelligent library services for university students. Mediterranean Journal of Basic and Applied Sciences, 9(1): 142–147.
- [56] Khan, N.A., Siddiqi, A.M.U., & Ahmad, M. (2021). Development of intelligent alumni management system for universities. Asian Journal of Basic Science & Research, 3(2): 51–60.
- [57] Khan, N.A., et al. (2019). Prevention of web-form spamming for cloud based applications: A proposed model. In 2019 Amity International Conference on Artificial Intelligence (AICAI), IEEE.
- [58] Kumar, D., & Dhir, S. (2021). Trends in Future Informatics and Emerging Technologies: Advanced Computing Techniques: Implementation, Informatics and Emerging Technologies, Volume 1.
- [59] Mahafdah, R.F., Al-Omari, O.M., & Khan, N.A. (2018). Learning modal adaptability to improve reading and writing skills of students.
- [60] Ren, M., et al. (2026). Explainable AI for structure-based drug design: Concepts and case studies. In Artificial Intelligence in Precision Drug Design, Volume 2, Pages 433–455, Elsevier.
- [61] Siddiqi, A.M.U., et al. (2026). Design and development. In Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025), Springer Nature.
- [62] Siddiqi, A.M.U., et al. (2025). Design and development of a cost-effective software maintenance management model. In International Conference on Innovative Computing and Cutting-edge Technologies, Springer Nature Switzerland Cham.
- [63] Zamani, A.S., Akhtar, M.M., & Khan, N.A. (2025). An application of machine learning, big data and IoT of enterprise architecture: Challenges, solutions and open issues. In Quality Control–Artificial Intelligence, Big Data, and New Trends, IntechOpen.
- [64] Zamani, A.S., et al. (2026). Challenges in docking accuracy and reproducibility. In Recent Advances in Molecular Docking, Pages 435–446, Academic Press.