IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

Acadion: AI-Based Academic Risk Prediction, Performance Analytics and Personalized Learning Support System

Mrs. Vedashree C R[1], Prashant[2], Renkigouda S H[3], Sadhana S[4], Santhosh Kumar S[5]

Assistant Professor, Dept. of CSE, Don Bosco Institute of Technology, Bengaluru, India[1]

Dept. of CSE, Don Bosco Institute of Technology Bengaluru, India[2-5]

Abstract: University faculty often diagnose weak students from raw marksheets that do not tie sub-question scores to curriculum topics, plan syllabus delivery by hand, and use risk indicators that give a label without a reason. This paper presents Acadion, a full-stack academic intelligence platform for these tasks. Acadion ingests Excel rosters and internal-assessment (IA) marks, maps scores to topics, and rolls them up through a four-level hierarchy that applies VTU IPCC scaling of IA marks from 40 to 15. A hybrid extractor, a heuristic parser with an LLM fallback, builds question blueprints from exam papers. A Random Forest assigns each student to ON_TRACK, NEEDS_ATTENTION or HIGH_RISK, a retrieval-augmented tutor answers topic questions from uploaded module notes with module citations, and a scheduling module plans teaching sessions against an academic calendar. The system comprises 177 REST endpoints over React, Spring Boot, MongoDB and FastAPI services. We report results candidly. The risk labels are produced by threshold rules over the same features the model sees, so the 100% test accuracy on 2,400 synthetic records shows that the model reproduces the rules, not that it predicts outcomes; a TreeSHAP analysis shows the scaled CIE score dominates (mean |SHAP| 0.232). The blueprint extractor recovered two of three plain-text test papers exactly, and on the third produced four spurious sub-questions. RAG answers carried citations in 10 of 10 test queries. Validation on labelled outcome data, scanned papers and real users is planned.

Index Terms: Academic Risk Stratification, Explainable AI, SHAP, Learning Analytics, Retrieval-Augmented Generation, Syllabus Scheduling, VTU CIE Scaling

I. INTRODUCTION

Universities record student performance as raw marks, yet a raw mark on a sub-question says little about which curriculum topic the student has not understood. Conventional learning management systems such as Moodle and Canvas manage content and communication but do not map assessment results to topics [1], and the faculty who use them still spend considerable time preparing question blueprints, schedules and marksheets by hand. Generative AI is widely expected to relieve this workload [2], but it needs safeguards before it can be trusted in a classroom.

Research on performance prediction has produced weekly, history-aware models built on LMS behaviour [3] and multi-classifier ensembles [4], and work on LLM-based question generation is growing quickly [5]. Three gaps remain for everyday institutional use. First, diagnostics stop at the marksheet and are not linked to curriculum topics. Second, most risk indicators output a label without saying why, whereas Shapley-value attributions [6] can expose the drivers of an individual prediction. Third, raw LLMs can hallucinate, whereas retrieval-augmented generation [7] anchors answers in source documents.

This paper presents Acadion, an academic intelligence platform for university faculty and undergraduate students, developed at Don Bosco Institute of Technology and aligned with VTU IPCC evaluation rules. It has three objectives. **O1 (risk analysis and explainability):** identify weak areas, stratify academic risk and explain it. **O2 (content studio and automated assessment):** parse exam papers and generate quizzes and slide decks to reduce preparation time. **O3 (academic planning and management):** use syllabi, teaching schemes, calendars and marksheets to support topic planning, syllabus tracking and multi-classroom management.

The contributions are: (i) a four-level diagnostic rollup from question-level marks to semester trajectory with VTU IA scaling; (ii) a hybrid heuristic-plus-LLM question-blueprint extractor; (iii) a Random Forest risk stratifier with an offline TreeSHAP analysis, together with an explicit account of what these results do and do not show; (iv) a hybrid dense-and-sparse retrieval tutor grounded per classroom; and (v) a syllabus scheduling module built into a multi-tenant classroom architecture.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

II. RELATED WORK

Wan et al. [3] proposed a two-stage transformer that fuses weekly LMS behaviour sequences with classroom head-pose features to predict a student's performance in the following week, reporting ROC-AUC of 81.02% and 82.65% on two blended-learning courses. The work shows the value of week-level, history-aware prediction, but it depends on classroom video and returns a prediction rather than a topic-level diagnosis. Acadion derives its features from marksheets that every institution already holds.

Awalurahman et al. [5] reviewed 28 primary studies on multiple-choice question generation with Transformers and LLMs. They observed that LLM prompting is increasingly used alongside fine-tuned Transformers, and that chain-of-thought and retrieval-augmented generation remain little used. Acadion applies retrieval augmentation to remediation rather than question writing, grounding each answer in the teacher's own module notes.

Butt et al. [4] combined bagging, boosting, voting and stacking classifiers to predict student performance in higher education and reported gains over single classifiers, but on a single institutional dataset and without an integrated academic-management workflow. Alserhan et al. [1] used structural equation modelling to show how LMS-based personal learning environments strengthen students' self-regulation; the focus is the learner, whereas Acadion also automates teacher-side diagnosis and planning. Ng and Ho [2] mapped generative-AI research in education through a bibliometric analysis of 3,808 articles. As a mapping study it delivers no system; Acadion turns these themes into working modules.

Table I: Summary of Related Work

Ref.	Approach	Reported result	Gap addressed
[3]	Two-stage transformer, LMS + head-pose fusion	ROC-AUC 81.02% / 82.65%	Needs classroom video; no topic-level feedback
[5]	Systematic review of LLM-based MCQ generation	28 studies reviewed	RAG little used; no platform
[4]	Multi-model ensemble for performance prediction	Gain over single models	One institution; no workflow
[1]	LMS-based personal learning environment, SEM	Self-regulation effect	Learner-centred; no teacher tools
[2]	Bibliometric map of GenAI in education	3,808 articles	Conceptual; no system

III. METHODOLOGY

A. System Architecture

Fig. 1 shows the architecture. The presentation tier is a React 19 single-page application built with Vite and Tailwind CSS, with teacher, student and admin views. The backend is a Java 17 Spring Boot 3.5 service with stateless JWT authentication (HS256) and the roles STUDENT, TEACHER and ADMIN, OTP-based registration (a six-digit code valid for 10 minutes, sent over SMTP), Excel parsing with Apache POI and PDF handling with PDFBox. It exposes 177 REST endpoints across 20 controllers. A Python FastAPI service hosts the risk model, the retrieval engine and the blueprint extractor and calls Gemini and OpenRouter models. A separate FastAPI service generates PowerPoint files. MongoDB Atlas stores all data.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

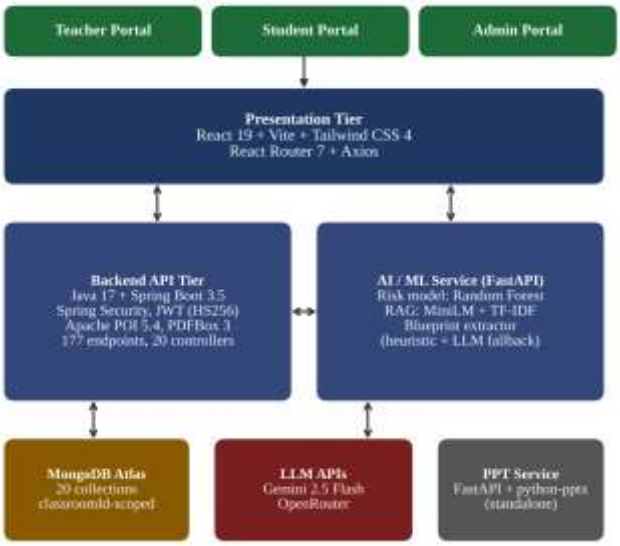


Fig. 1. Architecture of Acadion.

B. Multi-Tenant Classroom Isolation

Collections that hold classroom data carry a classroomId, and repository queries filter on it so that a request scoped to one course does not read another course's records. A teacher creates a classroom and shares a six-character join code, which is stored as a hash with an expiry time and an enabled flag. The platform holds 20 collections, including IA mark records, IA question blueprints, syllabus items, scheduled sessions, calendar events and a subject evaluation scheme that stores the scaling constants.

C. Roster and Marks Ingestion

Rosters and mark sheets (.xlsx and .xls) are read with Apache POI. Student rows are identified by the University Seat Number pattern (?i)\b([0-9][A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{3})\b, so headers and footers are skipped. A marks record stores a map from question code to score. The service also generates blank and master Excel templates from a confirmed blueprint so that marks from physical answer scripts can be entered offline and uploaded.

D. Question Blueprint Extraction

A blueprint is extracted in two stages (Fig. 2). The input may be a PDF, DOCX, plain text or an image passed through an OCR stage. Stage 1 is a heuristic state machine that applies regular expressions and layout rules, detects whether the paper is theory or multiple-choice, and emits each question with its code, maximum marks and topic text. Only if Stage 1 fails does Stage 2 call an LLM through OpenRouter with a zero-shot prompt that requests structured JSON. The confirmed blueprint is later joined with the marks to compute topic mastery. The extractor does not read the course-outcome or Bloom-level columns of the paper.

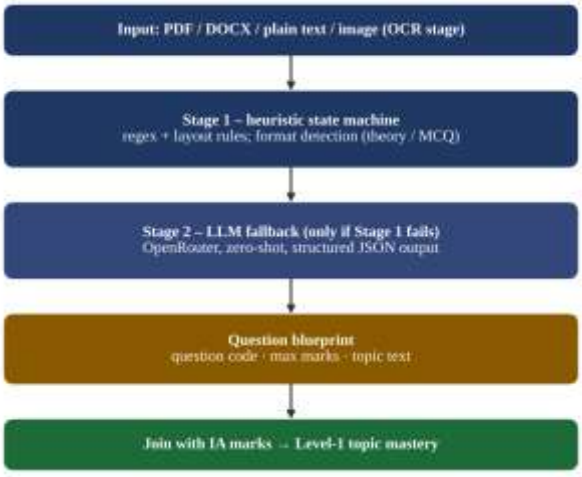


Fig. 2. Two-stage question blueprint extraction.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

E. Four-Level Diagnostic Rollup and VTU Scaling

Marks are rolled up through four levels (Fig. 3). Level 1 turns question-level marks into a score for each topic. Level 2 averages topics into a mastery value for each syllabus module. Level 3 computes the scaled continuous internal evaluation (CIE). Level 4 follows a student across successive IAs and subjects. At Level 3 the platform scales the average raw IA score as follows:

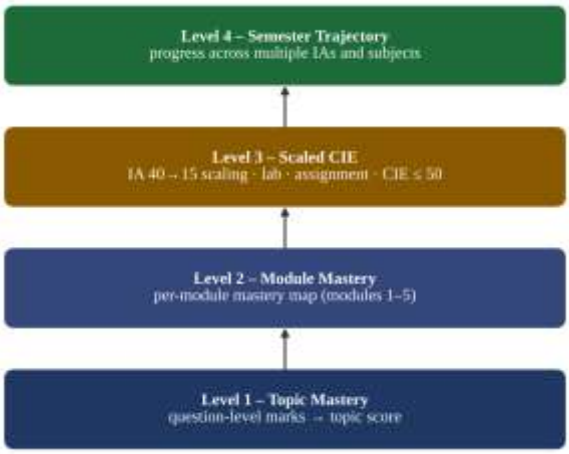


Fig. 3. Four-level diagnostic rollup.

$$S_{IA} = min(15, (R_{IA} / 40) \times 15) \quad (1)$$

$$CIE = S_{IA} + S_{Lab} + A \quad (2)$$

where R_{IA} is the average raw IA score out of 40, S_{Lab} is the scaled lab component and A is the assignment score (at most 10). The evaluation scheme stores a total CIE maximum of 50. For example, a raw score of 36.0 scales to 13.5 and a raw score of 38.0 scales to 14.25.

F. Risk Stratification and Attribution

Each student is described by five features (Table II). A Random Forest [8] with 100 trees, maximum depth 6 and seed 42 assigns one of three classes: ON_TRACK, NEEDS_ATTENTION or HIGH_RISK. The model is served at POST /ml/predict-student-risk. Training uses 2,400 synthetic records drawn from class-specific Gaussian clusters (target proportions 50%, 30% and 20%), and labels are assigned by explicit rules over the same features. HIGH_RISK applies if cie_scaled_pct < 40, or if it lies between 40 and 60 and either weak_topics_count ≥ 4 or module_mastery_avg < 30. NEEDS_ATTENTION applies in the remaining cases below 60, and at 60 or above when weak_topics_count ≥ 3 or module_mastery_avg < 50. All other students are ON_TRACK.

Table II: Risk Model Features

Feature	Definition
cie_scaled_pct	Scaled IA score as a percentage of 15
weak_topics_count	Topics with mastery below 40%
module_mastery_avg	Mean module mastery (0–100)
choice_skip_ratio	Share of choice groups where the harder option was skipped
attendance_pct	Attendance; defaults to 85.0 because no attendance module exists

For analysis, shap.TreeExplainer [9] gives each feature a Shapley value φ_i so that the model output decomposes additively:

$$f(x) = \varphi_0 + \Sigma \varphi_i \quad (3)$$

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

The attribution bars in the production interface are not computed this way. They come from hand-written formulas (for example, a term proportional to 60 minus the scaled CIE percentage) that are normalised to percentages. TreeSHAP was run only in the offline evaluation script (Section IV-D).

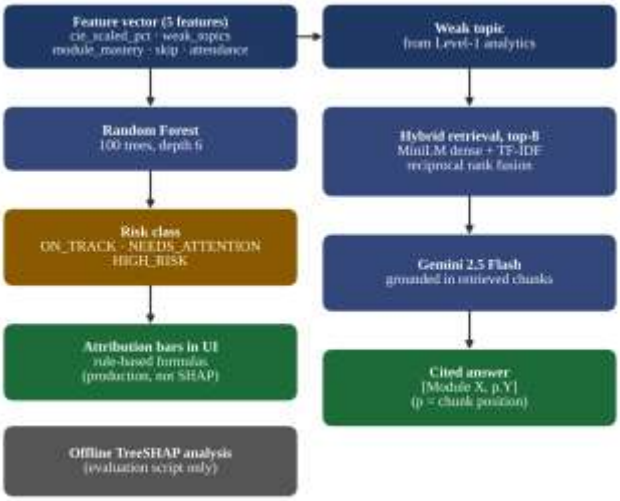


Fig. 4. Risk stratification, attribution and RAG flow.

G. Retrieval-Augmented Study Tutor

Uploaded module notes are split into chunks of about 900 to 1,200 characters (one paragraph, with about 150 characters of overlap) and embedded with all-MiniLM-L6-v2 [10] into 384-dimensional vectors, stored as flat NumPy files in a per-classroom folder. Retrieval combines dense cosine similarity with sparse TF-IDF scores using reciprocal rank fusion [11] and returns the top 8 chunks. Gemini 2.5 Flash writes the answer from those chunks and cites them as [Module X, p.Y], where Y is the chunk position in the uploaded document rather than a printed page number. The .env file names a newer model that returns HTTP 404, so the service falls back to Gemini 2.5 Flash.

H. Syllabus Scheduling and Slide Generation

The scheduling module (52 endpoints across four syllabus controllers, with 28 more for calendar, timetable and document handling) manages syllabus topics, an academic calendar, weekly timetable slots and teaching sessions, checks the feasibility of a plan against available hours, and publishes schedules with notifications. A separate FastAPI service uses python-pptx 1.0.2 to generate .pptx decks; it is not yet connected to the web interface.

I. Technology Stack

Table III: Technology Stack (Measured Versions)

Layer	Technology
Backend	Java 17 (Temurin), Spring Boot 3.5.14, Apache POI 5.4.1, PDFBox 3.0.5, jjwt 0.11.5
Frontend	React 19.2, React Router 7.15, Vite 8.0, Tailwind CSS 4.2, Axios 1.16
Database	MongoDB Atlas (database acadiondb)
ML / RAG	Python 3.14, FastAPI 0.136, scikit-learn 1.8, shap 0.54.1, sentence-transformers 5.4.1, PyMuPDF 1.27, pypdf 5.9
LLMs	Gemini 2.5 Flash; OpenRouter models for extraction fallback
Slides	python-pptx 1.0.2

IV. RESULTS AND DISCUSSION

Measurements ran on an Intel Core i7-9850H with 15.7 GB RAM under Windows 11 Pro. Three kinds of data are used and are labelled throughout: synthetic data for the risk classifier, three plain-text exam papers for extraction, and a pilot class of 66 real IA-1 records for ingestion and analytics, used with the class teacher's consent and without student names.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

A. Implementation Status

Table IV: Implementation Status of Modules

Module	Obj.	Status
Multi-tenant classrooms, hashed join codes	O3	Implemented
Excel ingestion and template generation	O1	Implemented
Four-level rollup, VTU scaling	O1	Implemented
Blueprint extraction	O2	Implemented; tested on text only
Random Forest risk model	O1	Implemented; synthetic training data
Attribution bars in UI	O1	Rule-based, not SHAP
RAG study tutor	O1	Implemented; one classroom indexed
Quiz generator	O2	Implemented; little use so far
Slide generator	O2	Standalone; not in the UI
Syllabus tracker and scheduler	O3	Implemented
Publish marks, OTP, JWT roles	—	Implemented

B. Pilot Data and Ingestion

The pilot class holds 69 real IA-1 records, collected with the class teacher's consent and analysed without names, of which 66 have non-zero totals. The scaled CIE percentage has mean 90.68, standard deviation 6.60 and range 72.5 to 100, and all 66 students fall in ON_TRACK. weak_topics_count and choice_skip_ratio are zero for every student because no choice-based blueprint was uploaded, and attendance_pct is fixed at 85. The pilot class therefore cannot test the risk model. The database also holds 2 classrooms, 86 syllabus topics, 80 scheduled sessions and 114 calendar events.

C. Risk Classifier on Synthetic Data

The 2,400 synthetic records split into 1,184 ON_TRACK, 733 NEEDS_ATTENTION and 483 HIGH_RISK, and a stratified 70/15/15 split gave 1,680 training, 360 validation and 360 test records. Table V reports the test results. The Random Forest scored 100% on every metric, with a diagonal confusion matrix (178, 110 and 72 correct), and five-fold cross-validation gave 1.000 ± 0.000 for accuracy and macro-F1.

Table V: Test Results on Synthetic Data (n = 360)

Model	Accuracy	Macro-F1	ROC-AUC
Random Forest	1.0000	1.0000	1.0000
Logistic Regression	0.9944	0.9923	1.0000
SVM (RBF)	0.9972	0.9962	1.0000
Decision Tree	1.0000	1.0000	1.0000
MLP (64-32, ReLU)	0.9972	0.9962	1.0000

These scores must not be read as predictive accuracy. Labels are computed by threshold rules over the same five features the model receives, so the task is to reproduce a noiseless function, and every baseline comes close to perfect. The ablation in Table VI confirms this: cie_scaled_pct alone reaches 100%, because the first rule tests only that feature. What the experiment does show is that a tree ensemble can encode the expert thresholds. Whether the thresholds predict real outcomes needs labelled data, such as semester-end results, that we do not yet have.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

Table VI: Feature Ablation (Synthetic Test Set)

Features	n	Acc.	Macro-F1	ROC-AUC
All five features	5	1.0000	1.0000	1.0000
weak_topics_count only	1	0.7611	0.7083	0.9099
module_mastery_avg only	1	0.9028	0.9042	0.9916
cie_scaled_pct only	1	1.0000	1.0000	1.0000
Attendance + skip ratio	2	0.9111	0.9058	0.9727
Without behavioural pair	3	1.0000	1.0000	1.0000

D. Attribution Analysis

Table VII ranks the features by mean absolute SHAP value over the test set. The scaled CIE score dominates, as the label rules predict. Fig. 5 shows the TreeSHAP values for one anonymised HIGH_RISK student (Student A, with cie_scaled_pct 36.92 and predicted probability 0.881): the low CIE score contributes most (+0.459), followed by the high skip ratio (+0.147).

Table VII: Global Feature Importance (Mean |SHAP|)

Rank	Feature	Mean |SHAP|
1	cie_scaled_pct	0.2322
2	module_mastery_avg	0.0927
3	weak_topics_count	0.0453
4	attendance_pct	0.0313
5	choice_skip_ratio	0.0270

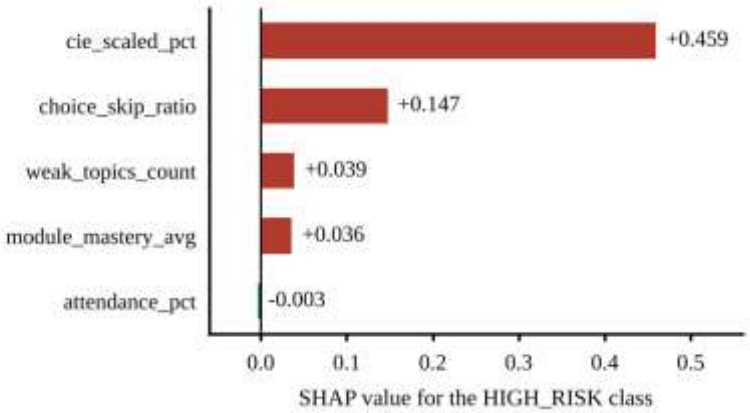


Fig. 5. TreeSHAP values for Student A (HIGH_RISK class, synthetic data).

E. Blueprint Extraction

Table VIII lists the three test papers, all supplied as plain text; no scanned PDF or image paper exists in the repository, so the OCR route is untested. The heuristic parser matched the question count on the two-module theory paper BIKS609 and the multiple-choice quiz 21CS54, with correct marks. On AI-INT-601 it returned 16 sub-questions for 12 expected, because two-line questions were split in two, and the marks of the split parts were wrong. Overall 28 of 32 extracted items (87.5%) matched an expected sub-question. Heuristic extraction took 3.4 ms on average. End-to-end calls to the LLM-backed /api/ai/parse-text endpoint averaged 21.8 s, and 3 of 10 calls timed out at 30 s.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131128

Table VIII: Blueprint Extraction on Plain-Text Papers

Paper	Format	Expected	Extracted	Time (ms)
AI-INT-601	Theory	12	16	8.1
BIKS609	Theory	12	12	1.1
21CS54	MCQ	4	4	1.1

F. Retrieval-Augmented Tutor

Ten topic questions were asked against the five indexed modules of one IoT course. Every answer carried citations (4 to 26 per answer) and was built from 8 to 13 retrieved sources. Mean latency was 9.69 s, or 32.83 s when a single 241 s outlier is included; that outlier was a model-name 404 followed by a retry. Without retrieved context the pipeline returns a fixed message that the topic is not covered in the notes, so no hallucination comparison with an ungrounded LLM was possible, and citation correctness was not checked by hand. In a separate timing run only 1 of 10 queries finished within 120 s for the same retry reason, which makes latency the weakest part of the tutor.

G. System Performance

Table IX: Endpoint Latency (Local Machine)

Endpoint	OK / n	Mean (ms)
Student analytics (four-level rollup)	10/10	162
IA marks record read	10/10	45
Class summary (69 students)	10/10	1,718
Risk and attribution (backend)	10/10	189
/ml/predict-student-risk	10/10	2,060
/api/ai/parse-text (LLM)	7/10	21,779
/rag/chat	1/10	12,234
Slide generation (cold 20.7 s)	5/5	8,283

The risk endpoint takes a near-constant 2.06 s because the service regenerates the synthetic data and refits the Random Forest on every request instead of loading a saved model. The frontend builds in 1.69 s into a 2.31 MB main bundle (634 kB gzipped) with no code splitting. The backend compiles, but of 257 unit tests 3 fail and 8 error, with 1 skipped; most errors come from an unconfigured mock in the roster-import tests. The code base has about 99,000 lines of Java, JavaScript and Python.

H. Limitations

The main limitations are these. (1) The risk labels are rule-derived, so the model has no independent validation. (2) The attribution bars shown to users are not SHAP values. (3) Attendance is a hard-coded default, because no attendance module exists. (4) The pilot class has almost no variance and one class only. (5) Extraction was tested on three text papers, mis-split two-line questions, and does not read course-outcome or Bloom columns. (6) RAG latency is unstable and only one classroom is indexed. (7) The slide generator is not integrated and the quiz generator has seen little use. (8) Several API routes still permit unauthenticated access and some defaults, such as the course code, are hard-coded. (9) No user study or manual baseline time was measured, so no time-saving claim is made.

V. CONCLUSION

This paper presented Acadion, a platform that links question-level marks to topics, applies VTU scaling, extracts blueprints, stratifies risk, tutors from course notes and schedules teaching. The working parts are the ingestion, rollup, scaling, scheduling and retrieval pipelines. The risk model currently shows only that a Random Forest can encode expert thresholds, and the user-facing explanations are not yet SHAP.

Next steps are to train and validate the classifier on labelled outcomes such as semester-end results, replace the rule-based bars with TreeSHAP and persist the trained model, add an attendance module, evaluate extraction on scanned

DOI: 10.17148/IARJSET.2026.131128

papers with course-outcome and Bloom fields, fix the model fallback that destabilises the tutor, connect the slide generator, and run a user study with measured time savings.

VI. ACKNOWLEDGMENT

The authors express sincere gratitude to Mrs. Vedashree C R, Assistant Professor, Department of Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru, for her continuous guidance, encouragement and valuable insights throughout the project. The authors also thank Dr. K B ShivaKumar, Professor and Head of the Department, and Dr. B S Nagabhushana, Principal, for providing the necessary facilities and support. Gratitude is also extended to Dr. Asha K H, Project Coordinator, and all faculty and staff who contributed to this work.

REFERENCES

[1]. S. Alserhan, T. M. Alqahtani, N. Yahaya, W. M. Al-Rahmi, and H. Abuhassna, "Personal learning environments: Modeling students' self-regulation enhancement through a learning management system platform," *IEEE Access*, vol. 11, pp. 5464–5482, 2023, doi: 10.1109/ACCESS.2023.3236504.

[2]. S.-L. Ng and C.-C. Ho, "Generative AI in education: Mapping the research landscape through bibliometric analysis," *Information*, vol. 16, no. 8, p. 657, 2025, doi: 10.3390/info16080657.

[3]. H. Wan, S. Yue, M. Li, X. Luo, Y. Hu, B. Che, and J. Wang, "Integrating blended learning behaviors via multimodal fusion for student performance prediction," *IEEE Trans. Learn. Technol.*, 2026, doi: 10.1109/TLT.2026.3660059.

[4]. N. A. Butt, Z. Mahmood, K. Shakeel, S. Alfarhood, M. Safran, and I. Ashraf, "Performance prediction of students in higher education using multi-model ensemble approach," *IEEE Access*, vol. 11, pp. 136091–136108, 2023, doi: 10.1109/ACCESS.2023.3336987.

[5]. H. W. Awalurahman, R. F. Aji, and I. Budi, "Transformer and large language models for automatic multiple-choice question generation: A systematic literature review," *IEEE Access*, vol. 13, pp. 127100–127112, 2025, doi: 10.1109/ACCESS.2025.3590423.

[6]. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 4765–4774.

[7]. P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.

[8]. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[9]. S. M. Lundberg *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56–67, Jan. 2020.

[10]. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[11]. G. V. Cormack, C. L. A. Clarke, and S. Büttcher, "Reciprocal rank fusion outperforms Condorcet and individual rank learning methods," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 758–759.