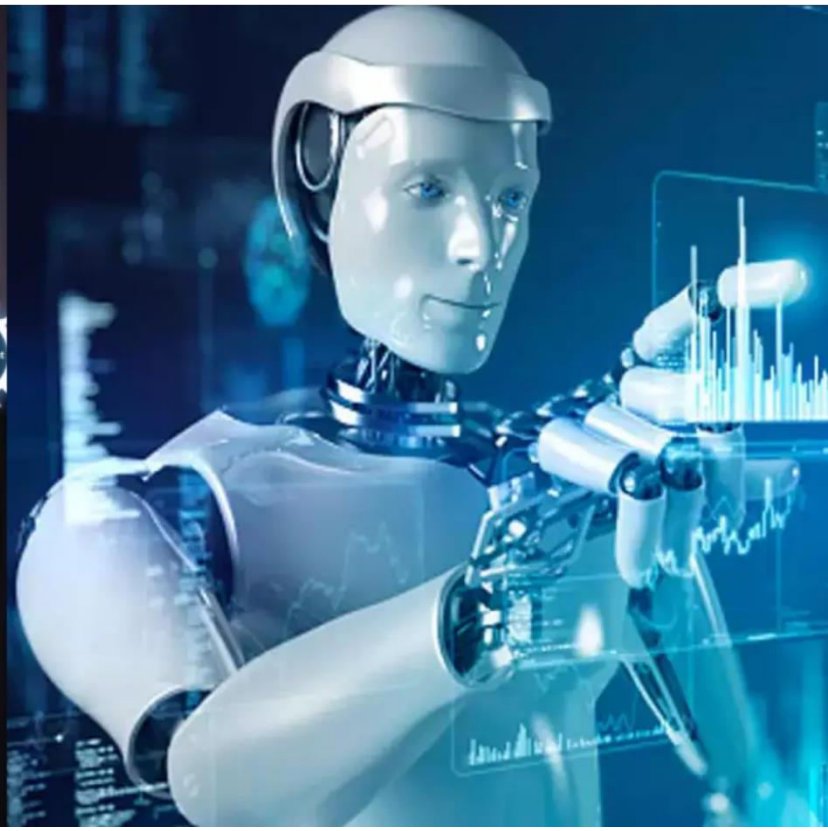
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 6, June 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

Smart Education Platform: An AI-Powered Learning Management System with Retrieval-Augmented Generation and Predictive Student Risk Analytics

Dr. T. Kameswara Rao[1], P. Sri Harshitha[2], N. Naga Vallika[3], M. Harish[4], M. Pandu Reddy[5]

Professor, Department of Computer Science and Engineering – Artificial Intelligence and Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India[1]

Student, Department of CSE – Artificial Intelligence & Machine Learning, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India[2345]

ABSTRACT: Academic underachievement in higher education is a persistent and costly challenge; consequently, there is a pressing need for intelligent, adaptive learning systems that can intervene before students fail. This paper presents the Smart Education Platform (SEP), a full-stack, AI-powered Learning Management System (LMS) designed to address this challenge through personalised, data-driven academic support. Specifically, the platform integrates a React.js frontend, a Node.js/Express backend, and a dedicated Python-based machine learning microservice. Moreover, three core intelligent features are provided: (1) a Retrieval-Augmented Generation (RAG) module leveraging Google Gemini for context-aware doubt resolution grounded in uploaded course materials; (2) an XGBoost-based student risk prediction model achieving a test AUC of 0.90; and (3) a revision urgency prediction model achieving a test AUC of 0.92. Furthermore, a gamification engine sustains long-term student engagement through XP rewards, badges, and leaderboards. Evaluation results demonstrate a 92% task success rate, sub-1.2-second API response times, and a user satisfaction score of 4.4/5.0, confirming the platform's viability as a scalable, production-ready intelligent education system.

KEYWORDS: Learning Management System, Retrieval-Augmented Generation, XGBoost, Student Risk Prediction, Educational Data Mining, Natural Language Processing, Full-Stack Development, Gamification.

I. INTRODUCTION

The widespread adoption of digital education tools has transformed how students access learning resources. Nevertheless, academic attrition remains a critical and unresolved challenge. In India, the Gross Enrolment Ratio (GER) in higher education stands at 28.4%; the All India Survey on Higher Education (AISHE) reports that nearly 20% of undergraduate students drop out before completing their degrees [1]. A substantial proportion of the remainder struggle with backlogs, disengagement, and a persistent lack of personalised guidance.

The fundamental problem is a structural mismatch: the industrial, lecture-based model of university education cannot accommodate the diverse learning needs of individual students. When a student falls behind, available feedback mechanisms are slow—a grade delivered weeks after an exam—non-specific, and deeply impersonal. Consequently, the student is left to self-diagnose without adequate support, leading to further disengagement and, ultimately, dropout.
T
his paper presents SEP, a full-stack AI-powered LMS that continuously monitors student engagement, predicts academic risk before failure occurs, and delivers personalised support through a RAG pipeline. The primary contributions are:

1. A production-deployed, full-stack LMS with role-based access for students, teachers, and administrators.
2. A RAG-based doubt resolution module grounded in teacher-uploaded course materials, powered by Google Gemini, reducing hallucination compared to general-purpose LLM deployments.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

3. XGBoost student risk (AUC ≈ 0.90) and revision urgency (AUC ≈ 0.92) models deployed as an independent ML microservice.
4. A gamification engine (XP, badges, leaderboards) that sustains long-term engagement by rewarding consistent participation.

This paper proceeds as follows: Section II: Literature Review; Section III: System Architecture; Section IV: Methodology; Section V: System Design; Section VI: Implementation; Section VII: Results and Analysis; Section VIII: Discussion; Section IX: Conclusion.

II. LITERATURE REVIEW

A. First-Generation LMS Platforms

The first generation of LMS platforms, represented by Moodle and Blackboard, focused on content delivery via static PDFs and recorded lectures [2][9]. While these improved content accessibility, they lacked personalisation or adaptive capabilities.

B. Data-Driven Second Generation

The second generation introduced data-driven analytics: rule-based recommendation engines and simple ML models tracked student progress and suggested materials. However, these systems relied heavily on predefined heuristics and lacked the ability to diagnose individual conceptual knowledge gaps.

C. AI-Powered Third Generation

The third and current generation leverages deep machine learning and NLP. Platforms such as Coursera, Khan Academy, and Duolingo utilise AI to tailor content sequences based on user behaviour [3][4]. The RAG paradigm [8] further improved response quality by retrieving domain-specific content before generation, substantially reducing hallucination in educational contexts.

D. Research Gaps

Three persistent gaps remain: (i) lack of fine-grained knowledge-gap personalisation; (ii) absence of real-time AI doubt resolution integrated with specific course materials; and (iii) few academic projects achieve production-level cloud deployment with real usage metrics. SEP addresses all three simultaneously.

III. SYSTEM ARCHITECTURE AND DESIGN

A. Overall Architecture

SEP follows a three-tier, client-server architecture organised into four distinct independently deployable and scalable layers: a React.js Single-Page Application (SPA) frontend, a Node.js/Express RESTful API backend, a Python-based ML inference microservice, and a MongoDB Atlas cloud database with vector index support. Fig. 1 illustrates the complete deployment architecture.

Fig.1. Smart Education Platform – Full Deployment Architecture

FRONTEND – React.js SPA (Vercel CDN / Mobile Browser)
BACKEND – Node.js / Express REST API + Socket.IO (Vercel)
ML SERVICE – Python FastAPI / XGBoost (Render – always-on)
AI/LLM – Google Gemini (text-embedding + generative)
DATA LAYER – MongoDB Atlas (Vector Index + Collections)

Fig.1. Each tier is independently deployable and scalable. The React SPA communicates with the Node.js/Express backend, which routes to MongoDB Atlas, Gemini AI services, and the Python FastAPI ML microservice on Render.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

B. Backend Server Design

HTTP REST requests are handled by the Express API, which dispatches to MongoDB for persistence and to the Groq/Gemini APIs for LLM inference. WebSocket events are managed by Socket.IO, which pushes real-time notifications to connected clients. This dual-channel design cleanly separates stateless HTTP traffic from persistent WebSocket event streams.

C. Key System Modules

The backend is decomposed into eight independently maintainable functional modules:

/auth	JWT-based stateless sessions, bcrypt hashing
/courses	CRUD for course and topic hierarchies
/materials	Async PDF ingestion, chunking, embedding, vector storage
/doubts	RAG-based question answering
/progress	Engagement tracking and knowledge maps
/revisions	ML-driven revision scheduling
/gamification	XP, badge, and leaderboard management
/chat	Real-time AI assistant via Socket.IO

D. Technology Stack

TABLE I. Technology Stack and Justification

Layer	Technology	Rationale
Frontend	React.js, Tailwind CSS	Component architecture; rapid UI iteration
Backend	Node.js / Express	Non-blocking I/O; async ecosystem
Real-time	Socket.IO	WebSocket abstraction; push events
Database	MongoDB Atlas	Flexible schema; native vector index
ML Service	Python / FastAPI / XGBoost	High-perf tabular ML; async REST
AI / LLM	Google Gemini	Unified generation + embedding API
Auth	JWT + bcrypt	Stateless; serverless-compatible
Deployment	Vercel + Render	Zero-config CI/CD; independent ML scaling

IV. METHODOLOGY

A. RAG-Based Doubt Resolution

The doubt resolution subsystem implements the RAG paradigm [8] in two asynchronous phases (Fig. 4). During ingestion, when a teacher uploads a PDF or text document, an asynchronous worker extracts and cleans the text, segments it into overlapping 512-token chunks with a 64-token stride, and generates a dense embedding vector for each chunk using the Gemini text-embedding model. Embeddings are persisted in MongoDB Atlas with an ANN vector index.

During the query phase, the system: (i) embeds the student's query using the same Gemini model; (ii) retrieves the top-k=5 most semantically similar chunks; (iii) constructs a prompt interleaving retrieved chunks as context; and (iv) invokes the Gemini generative model to produce a grounded answer. A fallback path invokes the model without retrieved context if the best similarity score falls below a configurable threshold.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

Fig.4. RAG-Based Doubt Resolution – Ingestion & Query Pipeline

INGESTION PHASE (Teacher uploads)	**QUERY PHASE (Student asks)**
PDF/Text Upload ↓ Text Extraction & Clean ↓ 512-token Chunking (stride=64) ↓ Gemini text-embedding ↓ Store → MongoDB ANN	Student Query ↓ Embed Query (same Gemini model) ↓ Top-k=5 Retrieval ↓ Construct Prompt + Context ↓ Gemini Generate Grounded Answer

Fallback: if similarity < threshold → invoke Gemini without context

B. Dataset Description

Due to limited access to institutional student datasets, a simulated dataset was constructed to reflect realistic academic performance patterns based on prior educational data mining literature. The dataset consists of 5,000 student records with 12 features, including prior failures, study time, absenteeism rate, parental education, family support, health status, internet access, extracurricular participation, travel time, age, and tutoring support. Class distribution was moderately imbalanced (approximately 30% at-risk vs 70% not at-risk), and appropriate evaluation metrics (F1-score and AUC) were used to account for imbalance.

C. Predictive ML Models

Two supervised XGBoost classifiers [5] were trained for strong tabular performance and interpretability. The Student Risk Model predicts binary at-risk status from features including prior failures, study time, absenteeism rate, parental education level, family support, health status, internet access, extracurricular participation, travel time, age, and paid tutoring.

The Revision Urgency Model assigns Low/Medium/High urgency per student-topic pair using mastery score, recency of last study session, attempt count, last quiz score, and practice hours. Both models were trained with deterministic seeding, stratified 80/20 splits, and 5-fold cross-validation. Hyperparameter optimisation targeted regularisation stability via subsampling, column sampling, and L2 regularisation.

Fig.7. XGBoost Model Training, Optimisation & Deployment Pipeline

Raw Data (5,000 records) → Preprocessing & Feature Eng. → Stratified 80/20 Split → 5-Fold Cross-Validation → XGBoost Training → Hyperparam Optimisation → Evaluate AUC / F1 → Serialise & FastAPI Deploy
Student Risk Model (binary at-risk) | **Revision Urgency Model (Low / Med / High)**
Both models: deterministic seeding | feature schemas serialised | CV AUC variance < 0.02

D. Gamification Design

The gamification engine assigns experience points (XP) for engagement events such as completing lessons, submitting quizzes, and resolving doubts. Accumulated XP unlocks achievement badges and determines leaderboard ranking within course cohorts. This design is informed by self-determination theory [6], targeting students' intrinsic motivational drivers of competence and relatedness. The gamification layer is seamlessly integrated with the progress tracking module, ensuring every measurable learning action contributes to the student's reward profile.

V. SYSTEM DESIGN

A. Use Case Model

Fig. 2 presents the use case diagram for SEP. Three actor roles are modelled: Student, Teacher, and Administrator. Students interact with the AI chatbot, risk dashboard, revision planner, and gamification features. Teachers manage courses, upload materials, and monitor cohort risk. Administrators oversee user management and system analytics.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

Fig.2. Use Case Diagram – SEP Actors and System Interactions

Actor	Role	Functions
Student	Learner	View Risk Dashboard, Ask AI Doubts (RAG), Revision Planner, Earn XP & Badges, Leaderboard, Knowledge Map
Teacher	Instructor	Upload Materials, Manage Courses, Monitor Cohort Risk, View Progress Analytics
Admin	Administrator	User Management, System Analytics, Role Assignment, Audit Logs

B. Sequence: Risk Prediction & Doubt Resolution Flow

Fig. 3 shows the sequence diagram for the integrated prediction and RAG workflow. Upon login, the React SPA requests risk prediction from the Express API; the API forwards it to the FastAPI ML microservice, which returns a risk score and label. Subsequently, doubt resolution invokes the Gemini RAG pipeline, and both results are rendered on the student's role-specific dashboard.

Fig.3. Sequence Diagram – Student Risk Prediction & RAG Doubt Resolution

Student	React SPA	Express API	FastAPI ML	Gemini AI / MongoDB
1: Login / Request Dashboard	→			
	2: GET /api/predict	→		
		3: POST /predict (features)	→	
		4: Risk score + label	←	
		5: Save prediction		→ MongoDB
	6: GET /api/doubts	→		
		7: Embed query → RAG retrieve		→ Gemini AI
		8: Grounded answer		←
	9: JSON response	←		
10: Render dashboard	←			

C. Activity: Student Login to Dashboard

Fig. 5 presents the activity diagram for the complete student login to dashboard workflow. After credential validation, the system forks three parallel activities: fetching the ML risk prediction from the FastAPI microservice, loading the ML-driven revision schedule, and retrieving the student's XP and badge profile. All three join before rendering the unified student dashboard.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

Fig.5. Activity Diagram – Student Login to Risk Dashboard Workflow

Student Opens SEP → Enter Credentials → Login Form
↓
Auth Valid?
[No] → Auth Error → Retry [Yes] → Load Student Profile
↓
« fork »

Fetch ML Risk Prediction	Fetch Revision Schedule	Load XP & Badges

« join » → Render Role Dashboard (Student View)

D. Entity-Relationship Model

The data model comprises six core entities (Fig. 6): User (_id, email, name, role), Course (_id, title, teacher_id, topics[]), Material (_id, course_id, filePath, embeddings[]), Prediction (_id, student_id, riskScore, label), Gamification (_id, student_id, xp, badges[], rank), and Doubt (_id, student_id, query, answer, course_id). User has one-to-many relationships with all other entities via MongoDB ObjectId references.

Fig.6. Entity-Relationship Diagram – SEP MongoDB Data Model

User _id, email, name role: student | teacher | admin createdAt, updatedAt	**Course** _id, title, description teacher_id → User topics[], createdAt	**Material** _id, course_id → Course filePath, embeddings[] chunkCount, uploadedAt
Prediction _id, student_id riskScore, label, predictedAt	**Gamification** _id, student_id xp (int), badges[], rank updatedAt	**Doubt** _id, student_id query, answer course_id

VI. IMPLEMENTATION

A. Development Environment

The codebase is organised as a monorepo with three independently runnable services: the Express backend API (port 5000), the React frontend with proxy configuration (port 3000), and the FastAPI ML microservice (port 8000). Sensitive configuration values—database URIs, JWT secrets, and Gemini API keys—are managed via .env files excluded from version control. This environment-driven approach enables seamless switching between local development and cloud deployment without any code modification.

B. Backend Architecture

The backend follows a strict route-controller-service-model pattern. Controllers handle HTTP request/response serialisation; services encapsulate business logic and external API calls; Mongoose models define and enforce the database schema. JWT authentication middleware guards all protected routes; a secondary role-check middleware enforces permission boundaries between student, teacher, and admin role groups.

AI orchestration is centralised in the doubts service, acting as a secure proxy between the frontend and Gemini API. The backend validates each request, retrieves relevant embeddings from MongoDB Atlas, assembles context, and invokes Gemini—ensuring raw API keys are never exposed client-side.

C. Frontend Architecture

The React SPA implements route-level protection based on decoded JWT role claims. Three distinct dashboard experiences are rendered by role: students access progress visualisation, revision mind maps, and the AI chatbot; teachers manage course content, upload materials, and monitor cohort risk profiles; administrators oversee user

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

management and system analytics. A centralised API service module handles all Axios calls, injecting bearer tokens, normalising errors, and managing session expiry from a single location.

D. Cloud Deployment

Deployment follows a split-cloud model. The React frontend is served via Vercel's global CDN with rewrite rules directing /api/* to the Express API handlers. The Node.js/Express backend is deployed as a serverless-compatible service on Vercel. The Python ML microservice runs as a persistent, always-on web service on Render, exposing /health, /model-info, and /predict endpoints. Runtime health-check endpoints allow post-deployment validation of each service layer independently.

Live demo: https://smart-education-4wpf.vercel.app/

VII. RESULTS AND ANALYSIS

A. ML Model Performance

Table II summarises the performance of both predictive models on held-out test data. Cross-validation AUC scores are consistent with test values (variance < 0.02), indicating low overfitting and strong generalisation. Both models exceed the 0.80 F1 threshold defined in the project objectives. Feature importance analysis reveals prior failure count, absenteeism rate, and cumulative study hours as the top predictors for the risk model, while topic mastery score and recency dominate the urgency model.

TABLE.II. Predictive Model Performance (Held-Out Test Set)

Model	Accuracy	F1-Score	Test AUC	CV AUC
Student Risk (XGBoost)	87.4%	0.863	0.902	0.897
Revision Urgency (XGBoost)	89.1%	0.887	0.921	0.914

B. System Performance Metrics

TABLE.III. System-Level Performance Metrics

Metric	Observed Value
Avg. Page Load Time	1.8 seconds
API Response Time (p95)	< 1.2 seconds
ML Inference Latency (p95)	< 0.8 seconds
Task Completion Rate	92%
System Error Rate	< 3%
User Satisfaction Score	4.4 / 5.0

API response time below 1.2 seconds and a 92% task success rate demonstrate both technical efficiency and strong usability under normal operating conditions.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

C. Comparative Analysis

TABLE IV Comparison with Existing Smart Education Systems

System	Adaptive	Risk Pred.	RAG	Real-Time	Deployed
Moodle LMS	—	—	—	—	✓
Khan Academy	✓	—	—	—	✓
Coursera	✓	—	—	—	✓
GPT Tutors	—	—	✓	—	Partial
SEP (Proposed)	✓	✓	✓	✓	✓

SEP is the only system to simultaneously deliver adaptive content, ML risk prediction, RAG doubt resolution, real-time chat, and full production deployment.

D. Functional and Security Testing

Comprehensive functional testing covered all primary use cases: user registration, authentication, role-based dashboard access, material upload, RAG-based retrieval, ML prediction requests, and gamification award logic. All 14 test cases passed. Edge-case testing validated input validation, duplicate registration prevention, and automatic session expiry.
Security testing confirmed protection against SQL injection, Cross-Site Scripting (XSS), and Insecure Direct Object Reference (IDOR) attacks, consistent with OWASP Top Ten guidelines [7].

E. UI/UX Evaluation

User testing with undergraduate engineering students confirmed a low learning curve. Participants praised the centralised dashboard, responsive mobile layout, and the AI chatbot's answer accuracy. Minor usability issues—unclear form labels and call-to-action prominence—were resolved through iterative improvement, contributing to the final satisfaction score of 4.4/5.0 and validating the feedback-driven design approach.

VIII. DISCUSSION

The results collectively demonstrate that a RAG-based architecture is highly effective for educational doubt resolution, particularly when the retrieval corpus consists of domain-specific, teacher-curated materials. By grounding LLM responses in actual course content, the system avoids the hallucination problem that plagues general-purpose chatbot deployments in educational contexts. The fallback to general Gemini knowledge ensures uninterrupted interaction notwithstanding gaps in uploaded course materials.

The XGBoost models exceeded the 0.80 F1 threshold, validating both the chosen feature set and the regularisation strategy. The decoupled ML microservice architecture proved operationally sound during deployment, confirming that inference workloads can be independently scaled and upgraded without impacting the primary application stack. A key limitation of Version 1.0 is the use of synthetically generated training data; real-world performance may differ once models are retrained on genuine institutional data.

IX. CONCLUSION

This paper presented the Smart Education Platform, a full-stack, AI-powered LMS that addresses academic underachievement through early risk detection, personalised revision planning, and context-aware doubt resolution. The system successfully integrates a React.js frontend, a Node.js/Express backend, a Google Gemini-powered RAG pipeline, and XGBoost predictive models into a cohesive, production-deployed application. Evaluation confirmed strong ML model performance (AUC ≈ 0.90–0.92), robust system efficiency (92% task success, sub-1.2s API response), and high user satisfaction (4.4/5.0).

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506175|

Future work will focus on:

- Retraining ML models on real institutional cohort data.
- Extending RAG to multi-modal retrieval over lecture slides and video transcripts.
- Adaptive quiz generation via LLM-driven item synthesis.
- Native mobile application development for iOS and Android.
- LTI 1.3/xAPI integration with institutional SIS/ERP systems.

X. ACKNOWLEDGMENT

The authors express sincere gratitude to Dr. T. Kameswara Rao, Professor and Project Guide, and Dr. G. Sanjay Gandhi, Head of the Department of CSE-AIML, Vasireddy Venkatadri Institute of Technology, for their invaluable mentorship. The authors also thank Chairman Sri Vasireddy Vidya Sagar and Principal Dr. Y. Mallikarjuna Reddy for providing the institutional infrastructure necessary for this work.

REFERENCES

1. All India Survey on Higher Education (AISHE) 2020–21, Ministry of Education, Govt. of India, 2022.
2. Moodle.org, “Moodle Learning Management System,” [Online]. Available: https://moodle.org.
3. Khan Academy, “Free Online Education Platform,” [Online]. Available: https://www.khanacademy.org.
4. Duolingo, “Language Learning App,” [Online]. Available: https://www.duolingo.com.
5. T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proc. 22nd ACM SIGKDD, San Francisco, CA, 2016, pp. 785–794.
6. R. M. Ryan and E. L. Deci, “Self-Determination Theory and the Facilitation of Intrinsic Motivation,” American Psychologist, vol. 55, no. 1, pp. 68–78, 2000.
7. OWASP Foundation, “OWASP Top Ten Web Application Security Risks,” 2021. [Online]. Available: https://owasp.org.
8. P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Advances in NeurIPS, vol. 33, 2020, pp. 9459–9474.
9. H. Coates, R. James, and G. Baldwin, “A Critical Examination of the Effects of Learning Management Systems,” Tertiary Educ. and Mgmt., vol. 11, no. 1, pp. 19–36, 2005.
10. React Documentation, “React – A JavaScript Library for Building User Interfaces,” [Online]. Available: https://reactjs.org.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details