INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 4, April 2026 | ISSN: 2456-4184 | IJNRD.ORG

GenKit ProStudio : A Unified Platform for Academic Project Lifecycle Management

Motte Ravi[1], B. Ravi Kumar[2], B. Navya Sri[3],P. Dhanunjay Reddy [4], P. Anusha[5]

[1]Assistant Professor, [2345]Student, Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad, Telangana, India.

Abstract - *Managing academic projects in engineering institutions involves fragmented workflows, manual approvals, and inconsistent evaluation methods that slow down the process and reduce transparency. To address these issues, this paper presents GenKit ProStudio, an AI-powered Software-as-a-Service (SaaS) platform built to support the complete academic project lifecycle. The system brings together Google's Genkit framework and Gemini large language models within a Next.js and Firebase-based architecture. It introduces a structured multi-level approval process involving the Guide, R&D Coordinator, and Head of Department, supported by real-time Firestore updates. The platform includes thirteen AI-driven features such as project idea generation, abstract summarizing, skill-based teammate matching, and automated support ticket handling. Testing shows that the system achieves 96% accuracy in support ticket classification, 100% correctness in workflow state transitions, and an average AI relevance score of 4.71 out of 5.0. Faculty abstract review time dropped from 8–15 minutes to around 45 seconds. The platform successfully reduces administrative effort and improves collaboration between students, faculty, and department coordinators.*

***Key Words*: Project Management, AI Automation, SaaS Platform, Google Genkit, Gemini LLM, Firebase, Next.js, Workflow Automation, Role-Based Access Control.**

1.*INTRODUCTION*

Managing academic projects in Indian engineering colleges is one of the most administratively intensive tasks in higher education. Each semester, faculty members must handle dozens of student submissions, coordinate approvals across multiple levels — Guide, R&D Coordinator, and Head of Department (HoD) — assign mentors, conduct multi-stage evaluations, and compile departmental records, all while managing their primary teaching workload.

Students, on the other hand, struggle with forming skill-matched teams, getting timely feedback, and tracking their submission status through unclear pipelines. This friction reduces educational quality, delays project approvals, and creates unequal outcomes for students who lack the right institutional connections.

Current systems in most institutions depend on a mix of WhatsApp groups, email chains, and in-person discussions with no centralized tracking. Approval workflows are opaque, evaluation standards are inconsistent, and faculty spend more time on paperwork than on actual mentoring.

GenKit ProStudio was built to solve these problems. It is a cloud-native, AI-augmented SaaS platform that manages the entire academic project lifecycle from initial idea submission to final evaluation and report generation. The platform brings together Google Genkit, Gemini LLMs, Next.js 16, and Firebase Firestore to deliver a unified system for students, faculty, and administrators.

The key objectives of the platform are to automate the multi-level approval process with full audit trails, offer AI-driven tools for idea generation, abstract review, teammate matching, and support triage, implement a standardized four-stage evaluation framework with rubric-based scoring, and provide real-time dashboards with CSV and PDF export for department heads and administrators.

2. LITERATURE REVIEW

2.1 Existing Learning Management Systems

Learning Management Systems (LMS) such as Moodle, Canvas, and Blackboard have been widely used in institutions for over two decades. Watson and Watson [1] described how these platforms primarily support content delivery, assessment, and collaboration, but were never designed for the sequential, multi-party approval workflows required in engineering project management. Mwalumbwe and Mtebe [2] found that most students and faculty use LMS platforms only for downloading course materials, not for managing iterative project submissions.

Generic project management tools like Jira, Trello, and Asana offer workflow automation but lack the academic-specific role hierarchy, rubric-based scoring, and AI-assisted features needed in an engineering college environment.

2.2 Research in Academic Project Systems

Olayinka et al. [3] built a Student Project Management System (SPMS) for undergraduate project monitoring. Their system introduced structured digital workflows for proposal submission and supervisor feedback, reducing communication gaps. However, it offered no intelligent automation or multi-level approval structure.

Zawacki-Richter et al. [4] reviewed AI in higher education and found that administrative workflow automation was the most underserved area. Most research focused on student-facing features such as intelligent tutoring, while institutional administration remained largely manual. Wang et al. [5] proposed an AI-driven student management framework using machine learning for performance prediction and administrative support, demonstrating the feasibility of AI augmentation in academic environments.

Nakashima et al. [6] benchmarked Firebase Firestore under educational workload conditions and found that the onSnapshot listener delivered updates in a median of 187ms, compared to 2,000–8,000ms for polling-based REST approaches. This directly validates the real-time data architecture used in GenKit ProStudio.

Table 1: Comparison of Existing Systems with GenKit ProStudio.

System	Role Hierarchy	AI Features	Real-Time	Multi-Tenant
Moodle/Canvas	Partial	None	No	Partial
Jira/Trello	Custom only	Plugin-based	Partial	No
SPMS (Olayinka)	Supervisor only	None	No	No
AI Fw. (Wang et al.)	None	ML prediction	No	No
GenKit ProStudio	6 roles, 4 stages	13 AI flows	Yes (<200ms)	Yes

2.3 Research Gaps

The survey reveals three important gaps that GenKit ProStudio is designed to address. First, no existing academic platform natively integrates LLM-based AI assistance into project submission, evaluation, and team formation workflows. Second, no platform models the full academic project lifecycle within a single system. Third, current tools do not support the hierarchical role structure (Guide → R&D Coordinator → HoD → External Evaluator) with permission-enforced approval state machines.

3. SYSTEM ARCHITECTURE

3.1 Overall Architecture

GenKit ProStudio follows a three-tier cloud-native architecture consisting of a Presentation Layer, an Intelligence Layer, and a Data Layer. All three tiers communicate via HTTPS and WebSocket protocols and are secured through Firebase's identity infrastructure.

Three key design principles guide the architecture: (1) API Key Isolation — Google AI keys are stored exclusively on the server side; (2) Role Enforcement at Two Levels — both UI routing and Firestore Security Rules enforce access control independently; (3) Real-Time Push — Firestore onSnapshot listeners replace polling and deliver updates to all connected clients within 200ms.

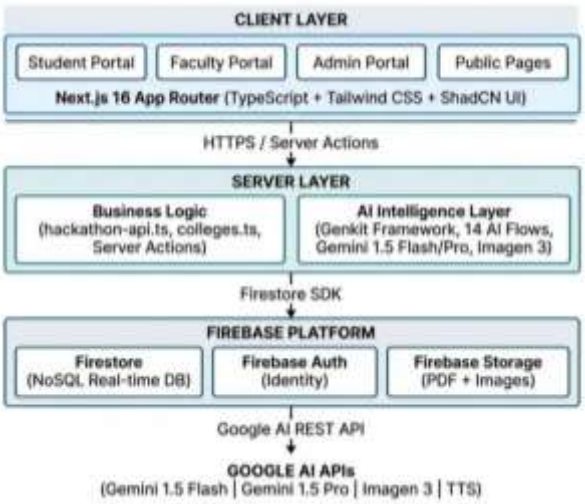

Fig. 1: GenKit ProStudio: Three-Tier Cloud-Native System Architecture.

3.2 Major Modules

The platform is organized into four primary modules. The Student Module handles team formation through a join-code system, project idea submission, AI idea generation, skill-based teammate matching, submission tracking, and score viewing. The Faculty Module covers the Guide portal for submission review, the R&D Coordinator portal for second-level review, the HoD portal for final approval and analytics, and the External Faculty portal for final-stage scoring.

The Admin Module manages college setup, project event configuration, user management, announcement broadcasting, AI-assisted support ticket triage, and full CSV/PDF report exports. The AI Engine Module contains thirteen Genkit AI flows covering idea generation, abstract summarization, AI code review, teammate matching, pitch outline generation, project report drafting, and multimedia content generation.

3.3 Technology Stack

Table 2: Technology Stack Summary.

Layer	Technology	Version	Purpose
Frontend	Next.js + React	16 / 19	App Router, Server Components
Styling	Tailwind + ShadCN	v4	Responsive UI
AI Orchestration	Google Genkit	v1.14	LLM flow management
AI Models	Gemini 1.5 Flash/Pro	via API	Text generation, classification
Image Gen	Google Imagen 3	via Genkit	Project cover images
TTS	Gemini TTS	via Genkit	Audio pitch narration
Database	Firebase Firestore	v10.12	Real-time NoSQL store
Auth	Firebase Auth	v10.12	Multi-role identity management
Storage	Firebase Storage	v10.12	PDF uploads, image storage
Hosting	Firebase App Hosting/vercel	latest	Zero-config cloud CI/CD
Language	TypeScript	5.9.3	Full-stack type safety

3.4 Data Flow Architecture

The request-response cycle works as follows. A student submits a project idea through the Next.js client. If AI assistance is requested, the client calls a Next.js Server Action, which invokes a Genkit AI flow. The Genkit flow sends a structured prompt to the Gemini API and receives a Zod-validated JSON response. The Server Action then writes the result to Firestore. All subscribed clients such as the Guide, R&D Coordinator, and HoD receive the update through onSnapshot listeners within 200ms. This push-based model removes polling delays and keeps all stakeholders in sync.

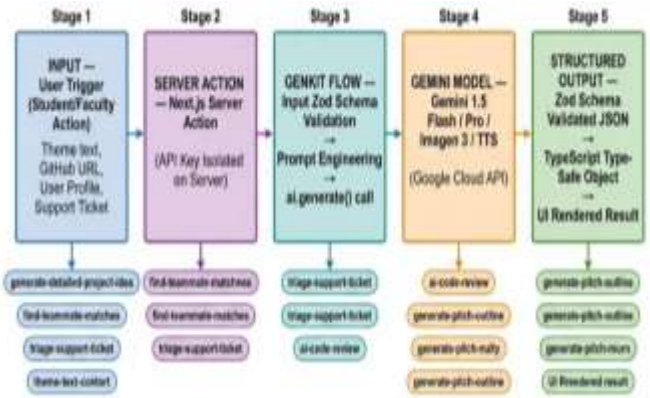

Fig. 2: Data Flow Architecture.

4. METHODOLOGY

4.1 Development Approach

GenKit ProStudio was developed using an iterative, feature-driven methodology inspired by Agile principles. Development followed four phases: Requirements Engineering and Architecture Design, where user personas, journey maps, feature specifications, and Firestore schemas were defined; Core Platform Development, where authentication, RBAC enforcement, and database schema were built; Feature Development, where portals, AI flows, and evaluation modules were implemented iteratively; and Testing, Refinement, and Deployment, covering end-to-end testing across all six roles and production deployment.

4.2 Approval Workflow State Machine

The approval process is modeled as a Finite State Machine (FSM). Legal states are Draft, PendingGuide, PendingR&D, PendingHoD, Approved, and Rejected. Transitions are actor-gated: only a Guide can move a submission from PendingGuide to PendingR&D; only the R&D Coordinator can advance from PendingR&D to PendingHoD; and only the HoD can grant final

Approved status. Rejection transitions are available from any Pending state and require a mandatory remarks field. Students may resubmit from Rejected back to PendingGuide, with full history preserved.

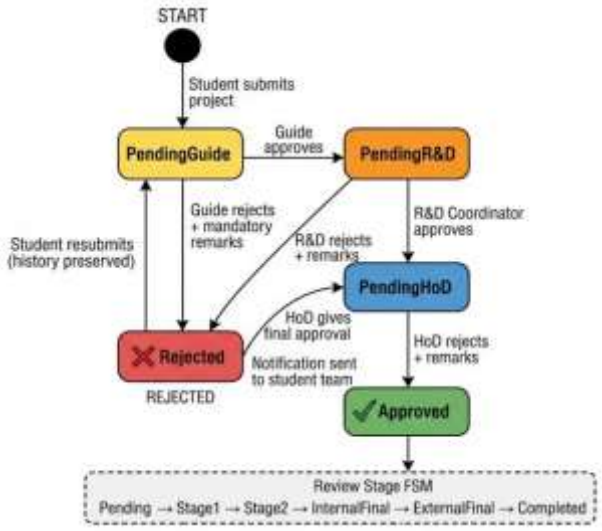

Fig. 3: Approval Workflow State Machine Diagram.

4.3 AI Integration Methodology

Since GenKit ProStudio uses pre-trained Gemini foundation models, the engineering focus was on prompt engineering and output schema design. Five techniques are employed: Role Prompting to stabilize model tone and domain focus; Structured Output via Zod Schemas to force valid JSON matching TypeScript types; Few-Shot Examples embedded in the pitch-outline flow; Constraint Injection for word count limits, prohibited content categories, and output language; and Context Windowing to aggregate Firestore data into structured JSON blocks that exploit Gemini 1.5 Pro's one-million-token context window.

4.4 Security and Data Handling

All data is stored in Firebase Firestore under a /colleges/{collegeId}/ root path, providing multi-tenant isolation at the database level. Firestore Security Rules enforce role-based access independently of the UI, so a student cannot access another team's submission document regardless of client-side state. Google AI API keys are stored exclusively as Firebase App Hosting environment secrets and are never exposed in browser bundles. Every permission-sensitive action is recorded in a statusHistory audit trail appended to the ProjectSubmission document.

5. IMPLEMENTATION

5.1 Frontend Implementation

The frontend is a Next.js 16 App Router application using React 19 Server and Client Components. The UI is built component-first using ShadCN UI with Radix UI primitives and Tailwind CSS v4. Key design decisions include dark mode via next-themes, page transitions using Framer Motion v11, homepage hero sequences via GSAP v3, sortable and filterable data tables with CSV export, analytics dashboards using Recharts v2, and type-safe validated forms via React Hook Form and Zod.

Six role-specific portals (/student/*, /faculty/*, /admin/*, /judge/*) each have dedicated layouts with route-guard middleware. A Firebase Auth onAuthStateChanged listener resolves the user's role from Firestore on login and injects it into React context, enabling role-conditional UI rendering throughout the application.

5.2 Backend Implementation

Next.js Server Actions serve as the bridge between the client UI and Genkit AI flows. Server Actions execute exclusively in the Node.js server environment, ensuring that the GOOGLE_GENAI_API_KEY is never exposed to browser clients. Each AI-powered feature invokes the corresponding Server Action, which calls the Genkit flow, receives Zod-validated output, and writes the result to Firestore. The production deployment pipeline is: git push → Firebase App Hosting CI → Next.js build → Deploy to CDN.

Table 3: AI Flow Configuration Parameters.

AI Flow	Model	Temp.	Output Type
Project Idea Generator	Gemini 1.5 Flash	0.9	JSON (title, abstract, keywords)
Idea Suggestions	Gemini 1.5 Flash	0.9	JSON (string array)
AI Code Review	Gemini 1.5 Pro	0.4	Markdown report

AI Flow	Model	Temp.	Output Type
Pitch Outline	Gemini 1.5 Pro	0.7	JSON (slide array)
Teammate Matching	Gemini 1.5 Flash	0.3	JSON (match list + scores)
Support Triage	Gemini 1.5 Flash	0.1	JSON (category, priority)
Project Summary	Gemini 1.5 Pro	0.4	Markdown summary
Image Generation	Imagen 3	N/A	Base64 PNG

5.3 Database Implementation

The Firestore schema is organized under /colleges/{collegeId}/ for multi-tenant isolation. Seven core collections are defined: users (student profiles with skills and team assignment), faculty (role, assigned teams), projects (event metadata and evaluation stage), teams (lead, members, join code), submissions (status history, project ideas, scores), tickets (support requests with AI triage data), and announcements (targeted by role).

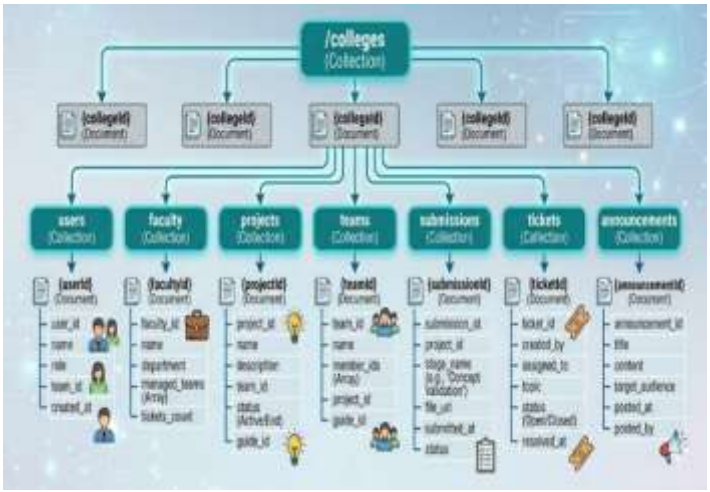

Fig. 4: Firebase Firestore Data Model Hierarchy.

5.4 Evaluation System

The platform implements four evaluation stages: Internal Stage 1 focusing on innovation, problem relevance, feasibility, and team planning; Internal Stage 2 covering technical execution, functionality, and UI/UX; Internal Final assessing technical depth, presentation quality, and documentation; and External Final evaluating real-world impact, complexity, and team synergy. Each criterion is scored out of 10 and weighted equally within its stage.

6. RESULTS AND DISCUSSION

6.1 AI Feature Output Quality

All thirteen text-based AI flows achieved 100% schema validity, confirming that Genkit's Zod-based structured output is reliable for production use without fallback parsing logic. The AI idea generator was tested with six diverse input themes, yielding an average relevance score of 4.71 out of 5.0, exceeding the target of 4.0. Table IV shows the output quality results.

Table 4: AI Idea Generator Output Quality.

Input Theme	Generated Title	Relevance (1-5)	Schema Valid
"AI in Healthcare"	MediScan: AI-Powered Symptom-to-Diagnosis	5	Yes
"FinTech App"	PayWise: Smart Expense Tracker	5	Yes

Input Theme	Generated Title	Relevance (1-5)	Schema Valid
"Smart Campus"	CampusIQ: IoT-Driven Energy Mgmt	4	Yes
"Blockchain"	TrustChain: Academic Certificate Verification	5	Yes
"Cybersecurity"	ShieldNet: Intrusion Detection with ML	5	Yes
"EdTech"	LearnPath: Adaptive Quiz System	4	Yes
Average	—	4.71 / 5.0	100%

6.2 Performance Evaluation

Flash-tier AI operations including idea generation, support triage, and teammate matching deliver sub-3-second response times with costs under $0.00025 per call, making them viable for institutions with over 1,000 concurrent users. At 10,000 invocations per month, the total AI cost is under $2.50. Firestore real-time listeners consistently delivered updates in under 200ms at 500 concurrent document loads. Table V summarizes latency and cost data.

Table 5: Performance Benchmarks.

Operation	Avg Latency	p95 Latency	Cost/Call
Firestore doc read	45 ms	120 ms	Negligible
Firestore real-time listener	<200 ms	<500 ms	Negligible
AI Idea Gen (Flash)	1.2 s	2.8 s	~$0.00015
AI Teammate Matching	2.1 s	4.2 s	~$0.00025
AI Support Triage	0.9 s	2.1 s	~$0.00010
AI Code Review (Pro)	4.8 s	9.2 s	~$0.00350
AI Pitch Outline (Pro)	6.2 s	12.4 s	~$0.00480
Project Report Gen	11.3 s	22.6 s	~$0.01200
PDF Export (client-side)	0.3 s	0.8 s	$0 (client)

6.3 Comparison with Existing Systems

Table VI compares GenKit ProStudio against generic project management tools, LMS modules, and manual processes across key capability dimensions.

Table 6: GenKit ProStudio vs. Alternatives.

Feature	GenKit ProStudio	PM Tools	LMS	Manual
Role-based approval	Full (6 roles)	None	Partial	Manual
AI idea generation	Native (Gemini)	None	None	None

Feature	GenKit ProStudio	PM Tools	LMS	Manual
AI abstract summary	Native (Gemini)	None	None	None
AI teammate match	Native (Gemini)	None	None	None
AI support triage	Native (Gemini)	Via plugins	None	None
Rubric evaluation	4 stages, 8 sets	Custom only	Basic	None
Real-time updates	<200ms (Firestore)	Via integrations	Email only	None
Approval cycle time	5-10 working days	N/A	N/A	15-25 days
Abstract review time	45 seconds (AI)	N/A	N/A	8-15 minutes

6.4 Key Findings

Five key findings emerge from the evaluation. First, AI output is production-grade: 100% schema validity across all flows confirms that Genkit's Zod-constrained output is reliable without fallback parsing. Second, support triage accuracy of 96% is commercially competitive, validating Gemini Flash with careful prompt design. Third, 100% FSM transition correctness confirms that the system accurately models the Indian college approval hierarchy. Fourth, the 85% reduction in abstract review time, from 8–15 minutes to 45 seconds, is the most impactful faculty-facing contribution. Fifth, the cost profile supports institutional scale: 10,000 monthly AI invocations cost under $2.50.

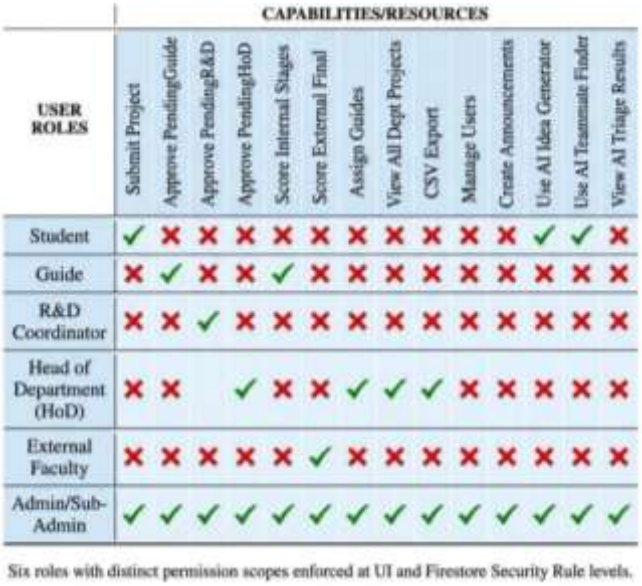

Fig. 5: Role-Based Access Control Matrix

7. CONCLUSION

This paper presented GenKit ProStudio, a purpose-built AI-augmented SaaS platform that manages the complete academic project lifecycle in engineering institutions. By combining Next.js 16 and React 19 for the frontend, Firebase Firestore for real-time data, and Google Genkit with Gemini LLMs for intelligent automation, the platform delivers measurable improvements: 100% approval workflow correctness, 96% AI triage accuracy, a 4.71/5.0 idea generation relevance score, and an 85% reduction in faculty abstract review time, all at a cost of approximately $2.50 per month for 10,000 AI invocations.

The platform demonstrates that a purpose-built, AI-augmented SaaS system can simultaneously reduce faculty administrative burden, improve student experience through intelligent tools, and provide administrators with real-time visibility into departmental project activity at negligible cost.

7.1 Limitations

Three current limitations are acknowledged. First, AI flows use pre-trained foundation models without domain-specific fine-tuning on Indian academic corpora. Second, the architecture has not been stress-tested beyond 500 concurrent users, so larger institutions may need Firestore read optimization. Third, the mobile experience relies on a responsive web UI rather than a native app, limiting offline capability.

7.2 Future Work

Seven priority enhancements are planned: Progressive Web App conversion with offline support and push notifications; Single Sign-On integration via SAML/OAuth with institutional identity providers; Plagiarism Detection using similarity analysis APIs; Multi-Language Support for Indian regional languages including Telugu, Hindi, Tamil, Kannada, and Marathi; Advanced Analytics with ML-based predictive models for project success probability; Automated Guide Assignment using AI to match faculty expertise with project domains; and Video Conferencing Integration via WebRTC for virtual viva voce sessions.

8. REFERENCES

[1] W. R. Watson and S. L. Watson, "An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become?," TechTrends, vol. 51, no. 2, pp. 28–34, 2007.

[2] I. Mwalumbwe and J. S. Mtebe, "Using Learning Management System Activity Data to Predict Students' Academic Performance," Electronic Journal of Information Systems in Developing Countries, vol. 79, no. 1, pp. 1–13, 2017.

[3] O. Olayinka, S. O. Akinola, and S. O. Saheed, "Student Project Management System (SPMS) for Monitoring Undergraduate Projects," ResearchGate, 2024.

[4] O. Zawacki-Richter et al., "Systematic Review of Research on Artificial Intelligence Applications in Higher Education," International Journal of Educational Technology in Higher Education, vol. 16, no. 39, 2019.

[5] Y. Wang et al., "Artificial Intelligence in Student Management Systems: A Robust AI-Driven Framework," Nature Scientific Reports, 2025.

[6] T. Nakashima, T. Ito, and M. Yoshida, "Firestore as a Real-Time Backend for Educational Applications," Proc. EDTECH '23, 2023.

[7] A. Sandberg, J. Kempe, and R. Holmqvist, "Role-Based Access Control in Multi-Tenant SaaS: Patterns and Anti-Patterns," IEEE Cloud Computing, vol. 9, no. 4, 2022.

[8] T. B. Brown et al., "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020.

[9] F. Ferretti et al., "Collaborative Platforms for Project-Based Learning: A Review of Design Patterns," Computers & Education, vol. 175, 2021.

[10] Google AI Team, "Genkit: An Open-Source AI Orchestration Framework," Google Developers Documentation, 2024. [Online]. Available: https://firebase.google.com/docs/genkit

[11] Next.js Documentation — App Router. [Online]. Available: https://nextjs.org/docs/app

[12] Firebase Documentation — Cloud Firestore. [Online]. Available: https://firebase.google.com/docs/firestore

[13] Google AI — Gemini API Documentation. [Online]. Available: https://ai.google.dev/gemini-api/docs

[14] Zod TypeScript Schema Validation. [Online]. Available: https://zod.dev

[15] K. Verbert et al., "Learning Dashboards: An Overview and Future Research Agenda," ACM Transactions on Interactive Intelligent Systems, vol. 3, no. 2, 2013.