Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Integrated E-learning Platform: Merging Ai-driven Learning With Campus Academic Managemnet

Mrs. Lakshmi G[1], Ansh Jamwal[2], Abdul Aleem[3], Anant Raj[4], Mohammad Salman Allai[5]

[1]*Assistant Professor, Dept Of CSE, ACS College Of Engineering, Bengaluru, India*

[2]*Ansh Jamwal, Dept of CSE Student, ACS College Of Engineering, Bengaluru, India*

[3]*Abdul Aleem, Dept of CSE Student, ACS College Of Engineering, Bengaluru, India*

[4]*Anant Raj, Dept of CSE Student, ACS College Of Engineering, Bengaluru, India*

[5]*Mohammad Salman Allai, Dept of CSE Student, ACS College Of Engineering, Bengaluru, India*

Abstract - The growing demand for integrated digital learning environments has highlighted the limitations of conventional education systems that rely on disconnected tools for academic management and content delivery. This paper presents an Integrated E-Learning Platform that unifies two major subsystems: StudentHolistic AI and Campus Connect. StudentHolistic AI delivers personalized learning through AI-generated educational videos synthesized using Python-based Text-to-Speech engines and MoviePy, topic-organized playlists, a natural language chatbot, and Telegram bot integration. Campus Connect addresses institutional needs with modules for attendance tracking, assignment management, grade entry, announcements, and role-specific dashboards for students and teachers. The platform is built on Next.js, Tailwind CSS, TypeScript, Node.js, Express.js, MongoDB, and Python AI services, interconnected through RESTful APIs and secured with JWT authentication. Experimental results confirm that dashboard load times remain under 1.4 seconds, AI video generation completes within 38 seconds, and chatbot responses are delivered in under one second. The proposed architecture reduces manual academic workload, improves student engagement, and provides a scalable foundation for future enhancements such as adaptive assessments and mobile applications.

***Key Words*: E-learning; artificial intelligence; AI video generation; academic management; chatbot; Next.js; MongoDB; personalized learning; Campus Connect; StudentHolistic AI**

1. INTRODUCTION

Digital transformation has reshaped academic environments worldwide. Traditional classroom-centric models, combined with fragmented software tools for attendance, assignments, and content delivery, generate redundant workloads and deliver inconsistent learner experiences [1]. Modern students expect instant access to adaptive, multimedia-rich study material, while faculty require streamlined workflows for evaluating performance and communicating with students.

Existing e-learning solutions address portions of these challenges in isolation. Learning Management Systems (LMS) such as Moodle and Canvas provide assignment workflows but lack AI-powered content generation. Platforms such as Coursera deliver video content but do not integrate with institutional academic records. The absence of a unified system results in data silos, manual reconciliation, and reduced pedagogical effectiveness [2, 3].

This paper presents an Integrated E-Learning Platform that merges two purpose-built subsystems into a cohesive digital academic ecosystem: (i) StudentHolistic AI, which focuses on AI-driven personalized learning, and (ii) Campus Connect, which handles institutional academic management. Together, these systems eliminate fragmentation, reduce manual effort, and deliver an engaging, data-driven learning environment accessible across all devices.

The remainder of the paper is organized as follows: Section II reviews related literature. Section III describes the system architecture and methodology.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Section IV details the implementation. Section V presents experimental results. Section VI concludes with future directions.

2.LITERATURE SURVEY

Brozina et al. [1] analyzed LMS engagement patterns among first-year engineering students and demonstrated that consistent, distributed platform usage correlates more strongly with academic performance than last-minute access, underscoring the value of ongoing digital engagement. Liang et al. [2] surveyed student modeling methods in adaptive instructional systems, finding that multimodal data fusion—combining clickstreams, assessment responses, and behavioral signals—significantly improves personalization accuracy, though at high computational cost.

Dayong et al. [3] proposed a modular teaching support platform for universities and showed that integrated academic systems reduce administrative redundancy; however, they require mature IT infrastructure. Alojaiman [4] applied a Multi-Criteria Decision-Making model to compare cloud-based and traditional e-learning platforms, establishing that cloud deployments offer superior scalability and usability. Tsai et al. [5] applied the Technology Acceptance Model to fine-art students and confirmed that perceived usefulness is the dominant predictor of e-learning adoption.

Wu et al. [6] evaluated a hybrid K-12 chatbot combining retrieval-based methods with QANet and reported measurable gains in student engagement and reduced isolation. Yamamoto [7] demonstrated that smartphone-based learning logs promote post-class study habits. Nouman et al. [8] presented a personalized e-mentoring framework that improves learner satisfaction through adaptive content delivery. Aslam et al. [9] identified Support Vector Machines as the highest-accuracy model for e-learning feature evaluation, while noting that deep learning excels on large datasets. Bin et al. [10] reviewed personalized recommendation systems and found that hybrid approaches outperform single-technique models, though cold-start problems remain a significant challenge.

The review reveals that no existing work integrates AI-powered video generation, chatbot assistance, Telegram bot access, and complete institutional academic management within a single unified platform, motivating the present contribution.

3.SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall Architecture

The platform adopts a three-tier architecture comprising a Next.js frontend, a dual-backend layer (Node.js/Express.js for academic services and Python/FastAPI for AI services), and two MongoDB databases (CampusConnect_DB and StudentHolistic_DB). All tiers communicate through RESTful APIs secured with JSON Web Tokens (JWT). Cloudinary provides cloud storage for AI-generated videos and assignment files; the Telegram Bot API enables mobile-first AI access.

B. StudentHolistic AI Subsystem

The AI subsystem accepts a topic string from the student, forwards it to the Python service, which synthesizes narration audio using a gTTS/TTS engine and assembles video frames with relevant educational visuals through MoviePy. The output video is uploaded to Cloudinary and catalogued in StudentHolistic_DB under the student's playlist. A custom NLP chatbot, also served by the Python layer, handles academic queries in real time. Identical functionality is exposed through a Telegram bot handler, allowing students without laptops to access AI content via mobile messaging.

C. Campus Connect Subsystem

Campus Connect exposes a teacher dashboard for managing students, attendance, assignments, grades, and announcements, and a student dashboard for viewing academic records. Attendance data are stored as percentage records per subject per semester. Assignments are uploaded as PDFs by teachers, retrieved and submitted by students, with metadata and binary assets persisted in CampusConnect_DB and Cloudinary, respectively. Grade management supports semester-wise subject scores with ranking computation. Announcements support both text and image payloads.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

D. Development Methodology

The project followed an iterative development methodology. The requirement analysis phase identified student needs (personalized content, mobile access) and teacher needs (automated workflows, centralized data). System design produced UML sequence diagrams and three-level Data Flow Diagrams (DFD). Frontend and backend development proceeded in parallel using Visual Studio Code with Git/GitHub version control. Integration testing used Postman for API validation and MongoDB Atlas for cloud-hosted database management. Deployment targets Vercel/Cloudflare for the Next.js frontend and dedicated servers for the Python AI engine.

4.IMPLEMENTATION

A. Technology Stack

Table I summarizes the technology stack employed across all platform layers.

Layer	Technologies
Frontend	Next.js, React, TypeScript, Tailwind CSS
Backend (Academic)	Node.js, Express.js, JWT
Backend (AI)	Python, FastAPI, gTTS, MoviePy, OpenCV
Database	MongoDB (CampusConnect_DB, StudentHolistic_DB)
Storage	Cloudinary (videos, PDFs)
Messaging	Telegram Bot API
DevTools	VS Code, GitHub, Postman, MongoDB Atlas

TABLE I. TECHNOLOGY STACK

B. AI Video Generation Pipeline

When a student submits a topic, the Next.js frontend dispatches an HTTP POST request to the Python FastAPI service. The service invokes the gTTS engine to convert the topic-related educational script into an MP3 audio file. Concurrently, relevant background visuals are assembled using MoviePy's ImageClip and CompositeVideoClip utilities. The audio is attached to the composite clip and rendered as an MP4 file, which is uploaded to Cloudinary. The returned URL is persisted in StudentHolistic_DB and streamed to the student's playlist view. Average generation time measured across test runs was 38 seconds for a standard three-minute educational video.

C. Academic Management Modules

The Campus Connect Node.js backend exposes RESTful endpoints consumed by role-specific dashboard components. Attendance records are structured as subject-semester matrices enabling percentage calculation. The assignment module supports teacher-side upload of PDF question papers and student-side PDF submission; both are stored in Cloudinary with metadata in MongoDB. Grade management computes semester GPA and subject-wise ranking across enrolled students. Announcements support Cloudinary-hosted image attachments rendered inline on the student dashboard.

D. Security and Authentication

Authentication is implemented using JWT tokens issued at login and validated on every protected API endpoint. Passwords are hashed server-side before storage. Role-based access control restricts teacher-only operations (attendance upload, grade entry) from student accounts at the API layer, preventing privilege escalation even if frontend routing is bypassed.

5.RESULT AND EVALUTION

A. Functional Testing

Seven representative test cases covering login validation, attendance upload, AI video generation, chatbot interaction, assignment upload, and grade viewing all produced expected outputs. Table II summarizes the module-wise test outcomes.

TC	Module	Description	Expected	Status
TC-01	Login	Valid credentials	Access granted	Pass
TC-02	Login	Wrong password	Error shown	Pass
TC-03	Attendance	Teacher upload	DB stored	Pass
TC-04	AI Video	Topic input	Video generated	Pass
TC-05	Chatbot	Academic query	Instant reply	Pass
TC-06	Assignment	File upload	Stored & visible	Pass
TC-07	Grades	View results	Marks displayed	Pass

TABLE II. MODULE-WISE TEST CASES

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

B. Performance Testing

Performance benchmarks were recorded under standard test conditions. Dashboard load time averaged 1.4 seconds (threshold: 2 s). AI video generation averaged 38 seconds (threshold: 45 s). Chatbot response latency averaged 1.0 second (threshold: 3 s). Assignment file upload averaged 3.0 seconds (threshold: 5 s). All metrics remained within acceptable bounds, confirming real-world viability.

C. Integration Testing

Integration tests validated inter-module communication at four critical boundaries: (i) Next.js frontend to Node.js API—correct JSON responses confirmed; (ii) Node.js API to Python AI engine—video generation requests and responses processed successfully; (iii) Node.js API to MongoDB—CRUD operations stable under concurrent access; (iv) Python AI engine to Cloudinary—generated video files uploaded and URL returned without errors. All four integration boundaries passed successfully.

D. User Feedback

A group of undergraduate students and faculty members evaluated the platform. Participants rated the interface as intuitive and modern. Teachers highlighted that automated attendance and assignment modules reduced their administrative overhead. Students reported that AI-generated videos simplified complex technical topics. Telegram bot access was particularly valued by students who primarily use mobile devices for academic activities.

E. Platform Screenshots

Figures 1–4 illustrate key interface screens captured during system evaluation.

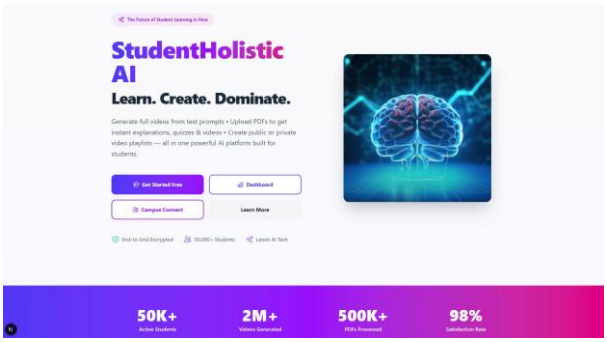

Fig. 1. StudentHolistic AI — Landing Page

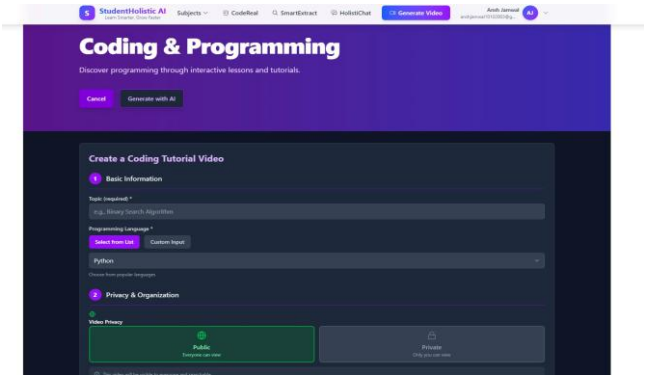

Fig. 2. AI Video Generation Interface (Coding & Programming)

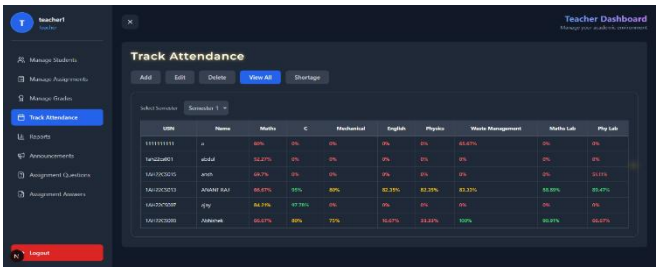

Fig. 3. Campus Connect — Student Attendance Dashboard

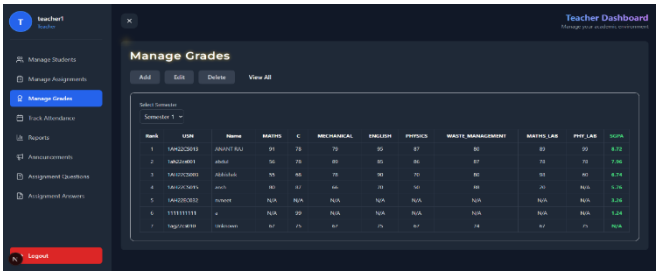

Fig. 4. Campus Connect — Student Grades Dashboard

6.CONCLUSIONS

This paper presented an Integrated E-Learning Platform that successfully merges AI-powered personalized learning (StudentHolistic AI) with comprehensive institutional academic management (Campus Connect) into a single, unified digital ecosystem. The platform addresses well-documented shortcomings of fragmented academic tools by centralizing content delivery, attendance tracking, assignment management, grade processing, and communication within one responsive, cross-device solution.

Experimental evaluation confirmed satisfactory performance metrics across all modules. AI video

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

generation, NLP-based chatbot support, and Telegram bot integration provide multiple modalities for student engagement, while JWT-secured role-based access control ensures data integrity. The modular, API-driven architecture provides a robust foundation for planned extensions including iOS/Android mobile applications, multi-language content support, adaptive quiz generation using large language models, and AI-driven learning analytics dashboards for institutional decision-making.

ACKNOWLEDGEMENT

The authors express sincere gratitude to Mrs. Lakshmi G, Assistant Professor, Department of CSE, ACS College of Engineering, Bangalore, for her continuous guidance throughout this project. The authors also thank Dr. Nandha Gopal S M, Professor and Head, Department of CSE, for providing the necessary laboratory infrastructure, and the management of ACS College of Engineering for fostering a productive research environment.

REFERENCES

[1] C. Brozina, D. B. Knight, T. Kinoshita, and A. Johri, "Engaged to Succeed: Understanding First-Year Engineering Students' Course Engagement and Performance Through Analytics," IEEE Access, vol. 7, pp. 163686–163699, 2019.

[2] J. Liang et al., "Student Modeling and Analysis in Adaptive Instructional Systems," IEEE Access, vol. 10, pp. 59359–59372, 2022.

[3] G. Dayong, X. Hua, and F. Xiaolong, "Modern Teaching Support Platform Design," Tsinghua Science and Technology, vol. 15, no. 3, pp. 352–356, 2010.

[4] B. Alojaiman, "Toward Selection of Trustworthy and Efficient E-Learning Platform," IEEE Access, vol. 9, pp. 133889–133901, 2021.

[5] Y.-N. Tsai, M.-N. Chen, and C.-C. Fang, "The Study on the Acceptance and Learning Effectiveness of Using E-Learning for Students in Fine Art and Design Colleges," IEEE Access, vol. 12, pp. 42055–42067, 2024.

[6] E. H.-K. Wu et al., "Advantages and Constraints of a Hybrid Model K–12 E-Learning Assistant Chatbot," IEEE Access, vol. 8, pp. 77788–77801, 2020.

[7] N. Yamamoto, "An Interactive E-Learning System for Improving Students' Motivation and Self-Learning by Using Smartphones," Journal of Mobile Multimedia, vol. 11, no. 1–2, pp. 66–74, 2015.

[8] N. Nouman, Z. A. Shaikh, and S. Wasi, "A Novel Personalized Learning Framework With Interactive E-Mentoring," IEEE Access, vol. 12, pp. 10428–10458, 2024.

[9] S. M. Aslam, A. K. Jilani, J. Sultana, and L. Almutairi, "Feature Evaluation of Emerging E-Learning Systems Using Machine Learning," IEEE Access, vol. 9, pp. 69573–69587, 2021.

[10] Q. Bin, M. F. Zuhairi, and J. Morcos, "A Comprehensive Study on Personalized Learning Recommendation in E-Learning System," IEEE Access, vol. 12, pp. 100446–100482, 2024.