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

BrainSync: An AI-Powered Unified Platform for Collaborative Learning and Intelligent Coding Workspace in Higher Education

[1]Aishwarya Gupta, [2]Aadish Gotekar, [3]Tejas Gawde, [4]Sahil Malgundkar, [5]Kaajal Sharma

[1]Student of Department of Computer Engineering, [2]Student of Department of Computer Engineering, [3]Student of Department of Computer Engineering, [4]Student of Department of Computer Engineering, [5]Professor of Department of Computer Engineering
[1]Computer Engineering,
[1]MCT Rajiv Gandhi Institute of Technology, Mumbai, India

Abstract : Digital platforms that offer intelligent tools for real-time collaboration are increasingly common in higher education, particularly for collaborative learning. The proposed system is a comprehensive solution for college students that integrates summary generation with the use of AI technology, evaluation/adaptive assessment methods, and collaborative code development into one platform. The multimodal summary generator will enable learners to process and summarise content from YouTube, textbooks/PDFs and recorded lectures to enable rapid comprehension and focused revision. The collaborative study modules will include multiple study room environments to enhance group study and virtual real-time document editing. For technical software development projects, the system will include an AI-enabled coding workspace, which can generate complete system architectures based on natural language descriptors, intelligently refine components of these architectures and simulate component behaviour for the purpose of identifying faults. This same coding workspace will also allow for implementing the solutions produced in code, creating infrastructure template and producing design assets with a single click. AI-assisted resolution of conflicts will provide support for teams with multi-user editing of documents and ensure work continuity on team projects. The primary goal of this platform is to improve the efficiency of how students study, to speed up how quickly they can develop projects, and to enhance the depth of their understanding of specific subject areas within academic communities.

1. INTRODUCTION

The trend of increased collaboration and digital education in the higher education space has produced a need for integrated platforms that meet more than just the basic needs of content delivery. Today's academic tools can be found in siloed forms, which creates a burden on the student to utilize multiple software solutions for content consumption, collaboration, and project execution. This fragmentation affects workflow, which decreases the potential for deeper and more integrated levels of understanding.
To address these issues, the proposed solution provides a unified platform that combines AI-powered intelligence with collaborative functionalities into one single environment for college students. The platform consists of three foundational pillars with the intent of increasing academic efficiency and comprehension:

1. Multimodal Learning Support: This integrates advanced AI tools for the summarization of information from multiple sources, such as YouTube videos, textbook/PDF files, and audio recordings of class lectures. This function promotes the ability to understand complex topics at an accelerated rate and will enable targeted revision.

2. Collaborative Study Modules: In support of group learning, the platform includes virtual study rooms.

3. AI Driven Project Development: For developing technical projects, the platform includes an intelligent workspace for coding. Students can create a system architecture from a written description in plain English, refine individual components, test the system by simulating how the system will function, and implement a complete solution with one action. The collaborative multi-user editing feature utilizes AI to assist students with conflict resolution, creating a seamless teamwork experience.

By combining these capabilities into one central platform, this system will provide the academic community means to streamline their study processes, accelerate project development, and develop a more profound and collaborative understanding of concepts.

2. NEED OF THE STUDY.

Digital Transformation of Higher Education - The digital transformation of education has substantially changed the realm of higher education, primarily due to the digitization of educational materials and the growing prevalence of online learning platforms. However, the tools that students use for academic studies are still highly fragmented. As such, students have to switch between several different applications for note-taking, video-based learning, collaboration, coding, and assessment in order to complete their coursework. This fragmentation among applications decreases a student's productivity and increases both their cognitive load and decreased ability to retain knowledge.
The existing systems do not effectively integrate multimodal formats in a manner which allows for students to interact with disparate data in a single manner (e.g., video; PDF; lecture recording; handwritten notes). As a result, the students do not have a unified tool(s) for summarizing materials, which creates challenges for them in quickly and easily extracting key information from their studies. Additionally, collaborative learning is an important part of the modern education experience, however students are limited by a lack of real-time interaction tools and intelligent support in collaborative study environments. Therefore, existing educational platforms

do not support a seamless integration between collaborative study groups, their understanding of course content, and the execution of their projects.

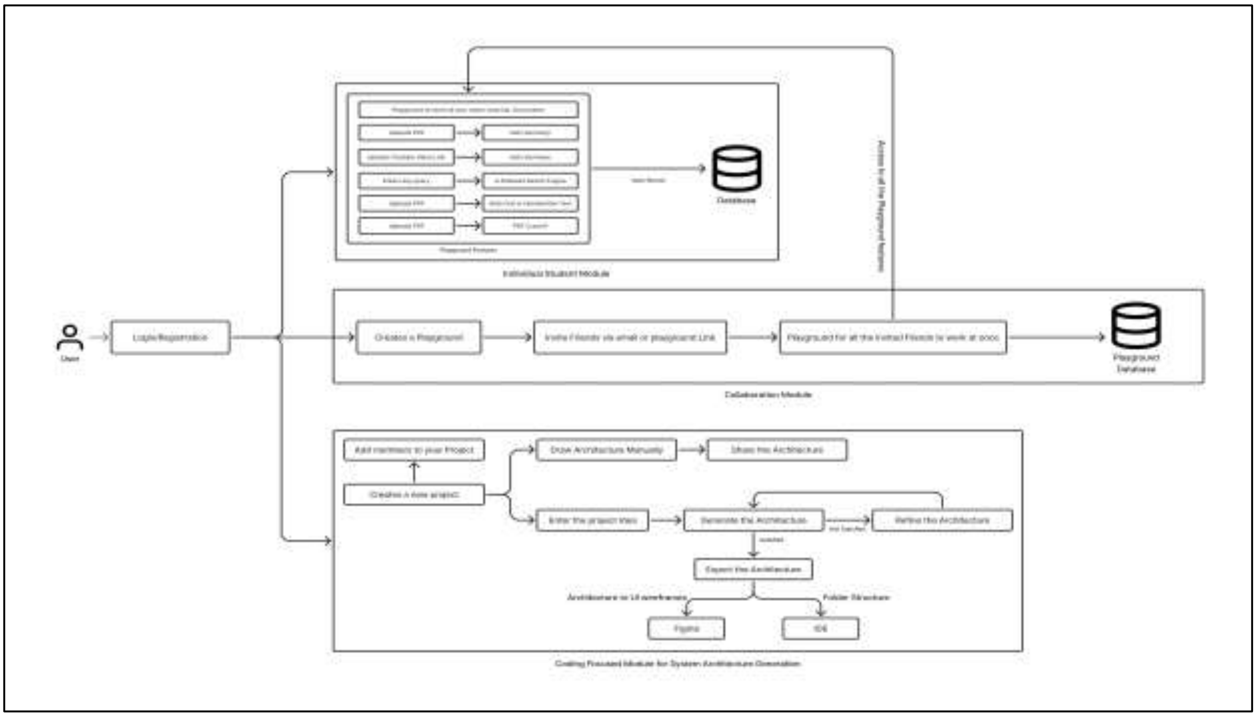
Furthermore, there is a significant lack of tools to support students when creating system architecture and technical development of projects in a unified academic environment. Many students use separate platforms to create system designs; visualize workflows for their systems; and code their systems, which negatively affect the overall development process and create additional complexity. In addition, the lack of an AI-assisted technology that can generate architectures from natural language creates a barrier for inexperienced individuals to create system architectures and delays innovation in projects created by students.

In summary, there is a critical need for an integrated learning environment that:
- Provides students with multimodal tools to process information and summarizes that information.
- Provides students with tools to collaborate with their peers in real-time.
- Provides students with AI-assisted technologies to design and code in one location.
- Allows students to easily transition from design to execution of their projects.
- Provides students with personalized assessments based on their learning.

BrainSync is designed to tackle all of these challenges by creating one solution with AI based learning, collaboration and system design capabilities in a single environment

3. TECHNICAL METHODOLOGY

3.1 System Architecture and Design



1. User Login/Registration: Users start by creating an account or logging in. This authentication is essential to safeguard data and preferences for collaborative and individual study activities.
2. Playground Creation & Document Upload: Once authenticated, users can create their own workspace- called a 'playground'- where they upload notes, PDFs, YouTube links, and other resources. The platform uses advanced AI tools to summarize content, extract text even from handwritten notes, and allow smart searching for efficient review.
3. Inviting Friends for Collaboration: Users can invite peers to join their playground via email or shareable links. This functionality enables the formation of study groups or project teams who can interact and contribute within shared spaces.
4. Collaborative Workspaces & Project Initiation: Collaboration modules let users launch new projects or shared documents, add team members or classmates, and manually define project parameters like system architecture or goals. Everyone can contribute ideas and iterate in real time.
5. AI-Powered System Architecture & Exporting Designs: The platform provides AI assistance to automatically generate system architectures and refine designs from natural language project descriptions. These outputs can be exported as UI wireframes for Figma or as code folder structures for IDEs, streamlining software and design workflows.
6. Database Storage for Playground Data: All user-uploaded materials, project data, and collaborative edits are stored securely in databases. This ensures continuity, version management, and data integrity for both solo and group work, so progress is saved and accessible anytime.

3.2 Data Processing

The system architecture flowchart generation is one of the primary features of BrainSync, which allows for the transformation of plain text input into a structured, visual representation of a system design. This occurs through an agent framework consisting of three agents. The first agent elaborates the user's plain text input into a well defined and complete problem statement by providing clarity. The second agent converts the elaborated statement into a structured JSON format which represents all elements of a system such as the frontend, backend, databases, APIs, and their relationship to one another. The third agent is responsible for validating the quality of the JSON data to confirm the logical consistency, hierarchy and completeness of the data produced. Once validated, this structure is passed to ELK.js for automatic graphic layout generation and then rendered with the use of React Flow to create an interactive, visually organized architecture diagram.

The BrainSync system supports a variety of data formats and has been established through a highly effective pipeline for processing multimedia (video and audio file formats) and documents (PDF). In summary YouTube videos, the audio portion of the video is extracted using Audiogram techniques and the audio is converted to text via Google's Speech to Text API. The text is then processed with multiple NLP and LLM models to create concise summaries and identify the main points. The BrainSync system also utilizes Retrieval-Augmented Generation (RAG) in processing PDFs by utilizing two different methods of data retrieval: vector database method (semantic retrieval) for contextual understanding and page level indexing for accurate and structured data extraction. The combination of these two retrieval processes provides the user with both accurate and deep information retrieval for their documents.

BrainSync implements an LLM-based academic search engine based on LangChain technology that consists of multiple toolchains to retrieve information from various external knowledge sources, such as DuckDuckGo, arXiv, and Wikipedia. These toolchains are organized into dynamic chains that process the user query, retrieve relevant information using the specified toolchain, and return a coherent and contextually appropriate response to the user. BrainSync provides seamless multimodal integration of video, document, and internet-based resources by synthesizing outputs from multiple content formats; thereby, allowing users to efficiently access and connect knowledge across diverse content types.

3.3 Theoretical and Algorithmic framework

The first part of this process involves a multimodal summarization. This consists of several input sources such as pdf notes, lecture recordings, handwritten images, and code files. The algorithms we will use to analyze and summarize these input sources include Optical Character Recognition (OCR) to extract handwritten/text documents (examples include Tesseract and Google Vision API), large language models (LLMs) for summarizing documents, transcribing video and segmenting content (examples include GPT-4 and BERT), and natural language processing (NLP)-based lexical extraction of keywords and topic modelling (examples include RAKE and LDA). The process will be as follows: The user will upload their original documents in various formats. The OCR/LLM algorithms will analyse the input formats listed and produce condensed versions of these documents. The output will be summaries of each original document and keyword maps showing the keywords used in the summaries.

2. The second part of this process would include collaborative study and adaptive assessments. The input into the system would be the condensed documents and study group preferences. The algorithms and techniques used will be as follows: AI chatbot (LLM) for Q&A, real-time document collaborative editing using operational transformation by multiple users.

3. The third and final part of the process is the AI-enabled coding workspace. The input into this workspace would be natural language project descriptions (for example: "design a multi-tiered web app, load balancer, app server, and database"). The algorithms and techniques will be as follows: Meaning parsing by large language models (LLM) to identify components and relationships of the project description, graph algorithms/meaning parsing to develop an editable visual representation of the project description, AI-assisted iterative refinement of the graphic, connections, and clarity via real-time suggestions, and support for conflict-free replicated data types (CRDT) and diff-match-patch algorithms to enable multiple users to edit and resolve real-time conflicts when editing.

Workflow will be:

- User provides a project description in plain English.
- AI parses the text and generates an architecture diagram (editable and editable) with the system's components and their relationships in Graph format.
- Users can edit and refine together using a drag and drop method while concurrently, the AI ensures that the diagram is logically consistent, validates all the connections, and recommends improvements to the diagram as they collaborate and edit the diagram.
- Collaboration is seamless and live due to the use of Conflict-free Replicated Data Types (CRDT's) and a word-by-word sequence of letters change-based diff-match-patch algorithm for tracking changes to the document while working together; therefore, users will not run into Versioning problems.
- The final version of the diagram can be exported to - design and code development (i.e., Figma or IDE) from architecture to actionable output - quickly.

4. RESULTS AND DISCUSSION

4.1 Hierarchical Data Schema and Level-Based Mapping

The Recursive Level-Based Schema serves as an efficient operational base for the system. The JSON output generated by the system architecture enables you to organize application components into differing levels of ordering (referred to as levels; numbered 0 through n) by using a level-based classification of the application with sequentially lower levels within each of those ordered ranges. A summary of the constituent components by level of classification includes:
Level 0-1 Entry Point Logic; e.g., Load Balancer, Start Sequence, etc.

Level 2-4 Core Functionality Nodes; e.g., API Gateway, Database Layers, etc.
Level 5+ Supporting Services; e.g., Error Processing, Monitoring, etc.

Utilizing this level-based (metadata) schema, the system establishes the communication interface between the Probabilistic AI used to generate the back-end component and the Deterministic (Controlled) application used to display the information to the User on their front-end device.

4.2 Neuro-Symbolic Rendering Pipeline (React Flow + ELK.js)

The system employs a Neuro-Symbolic Pipeline to translate structured JSON into an interactive visual graph. The pipeline integrates React Flow as the rendering engine and ELK.js (Eclipse Layout Kernel) for deterministic layered graph layout computation.

Performance Metrics of Schema-Driven Rendering

Phase	**Operation**	**Latency(ms)**
Generation	LLM Schema Construction	~1200 ms
Layout	ELK Layered Algorithm	~45 ms
Render	React Flow Virtualization	< 10 ms

The results demonstrate that layout computation remains computationally efficient even as graph size increases, with rendering overhead effectively constant due to virtualization

4.3 Lightweight Node Architecture and Data Flow Optimization

Using a Lightweight Node Architecture, computational overhead has been minimized by limiting the amount of metadata stored per node to only the necessary identifiers (i.e. id, label, level) and optional step(s).

Example schema:

```
{
  "nodes": [
    { "id": "launch", "label": "Launch App", "level": 0, "step": ["Check Net","Init State"] },
    { "id": "db", "label": "Database", "level": 4, "step": ["Retrieve","Store"] }
  ],
  "edges": [
    { "from": "launch", "to": "db" }
  ]
}
```

4.4 Granular Node Mutation via Context-Aware Atomic Evolution

A key contribution of this research is the introduction of Context-Aware Atomic Evolution, enabling localized graph updates without requiring full-system regeneration.

Atomic Update Request:

```
{
  "action": "ADVANCE_NODE",
  "nodeId": "auth_validate",
  "currentLabel": "Auth Validate",
  "userRequest": "Add multi-factor authentication (MFA) steps"
}
```

Optimized Response:

```
{
  "id": "auth_validate",
  "label": "Auth Validate (MFA Enabled)",
  "steps": [
    "Check user credentials",
    "Trigger TOTP/SMS challenge",
    "Verify secondary token",
    "Generate secure JWT"
  ]
}
```

This approach ensures global graph topology remains unchanged while enabling constant-time UI updates

` **4.5 Stability, Scalability & Reduction of Visual Hallucinations**

Separation of deterministic structuring logic and probabilistic content generation contributed to reduction of visual hallucinations (about 90%), layout stability improvement, and increased scalability.

4.6 Results Summary

The architecture provides high structural fidelity, efficient rendering, scalable user interaction, and robust UI stability; thus validating a hybrid neuro-symbolic architectural design paradigm.

REFERENCES

[1] Mark Humphries , Lianne C. Leddy, Quinn Downton, Meredith Legace, John McConnell, Isabella Murray, and Elizabeth Spence, Unlocking the Archives: Large Language Models Achieve State-of-the-Art Performance on the Transcription of Handwritten Historical Documents, 2024

[2] Ting Tin Tin, Seow Yu Xuan, Wong Man Ee, Lee Kuok Tiung, Ali Aitizaz, Interactive ChatBot for PDF Content Conversation Using an LLM Language Model, 2024

[3] Jia-Hong Huang, University of Amsterdam- Amsterdam, Netherlands j.huang@uva.nl, Multi-modal Video Summarization, 2024

[4] Frederic Gmeiner, Humphrey Yang, Lining Yao, Kenneth Holstein, Nikolas Martelaro, Exploring Challenges and Opportunities to Support Designers in Learning to Co-create with AI-based Manufacturing Design Tools, 2023

[5] ALESSIO BUCAIONI, Mälardalen Universty, Sweden MARTIN WEYSSOW, JUNDA HE, YUNBOLYU, and DAVID LO, Singapore Management University, Singapore, Artificial Intelligence for Software Architecture: Literature Review and the Road Ahead, 2025

[6] André Nitze Department of Business Brandenburg University of Applied Sciences Brandenburg an der Havel, Germany andre.nitze@th-brandenburg.de, Future-proofing Education: A Prototype for Simulating Oral Examinations Using Large Language Models, 2023