

# Connect-Ed: A Collaborative Learning Workspace

**Dhrupesh Savadia (Mentor)<sup>1</sup>, Saachi Dhumal<sup>2</sup>, Devansh Vadgama<sup>3</sup>, Daksh Jain<sup>4</sup>, Rudraa Hate<sup>5</sup>**

<sup>1</sup>Mentor, Department of Computer Engineering, Thakur Polytechnic, Kandivali East-400101

<sup>2</sup>Diploma Student, Department of Computer Engineering, Thakur Polytechnic, Kandivali East-400101

<sup>3</sup>Diploma Student, Department of Computer Engineering, Thakur Polytechnic, Kandivali East-400101

<sup>4</sup>Diploma Student, Department of Computer Engineering, Thakur Polytechnic, Kandivali East-400101

<sup>5</sup>Diploma Student, Department of Computer Engineering, Thakur Polytechnic, Kandivali East-400101

\*\*\*

**Abstract** - For note-taking, coding practice, document viewing, and online discussion, modern students often use several digital tools. This leads to fragmented learning processes and scattered information. Constantly switching between different platforms makes it hard for students to organize, review, and reuse information effectively, which decreases their productivity. ConnectEd is proposed as an integrated collaborative learning space that brings together various academic and development tools into one environment to address this issue. To promote interactive and collaborative learning, the platform includes visual knowledge boards, embedded code execution, structured note creation, and a community knowledge-sharing system. Users can create block-based notes, run programming code directly in the notes, visually organize concepts using boards, and share resources through a community platform designed for questions, code sharing, and note sharing. A document ingestion system is an additional feature that allows users to upload educational materials such as PDFs, presentations, and documents. These materials can be processed automatically and converted into structured notes for easier study and reuse. ConnectEd aims to simplify digital learning processes, improve knowledge organization, and create a more cohesive and effective educational environment for today's learners by integrating knowledge creation, experimentation, visualization, and collaboration into a single space.

**Key Words:** Collaborative learning, Structured note-taking, Embedded code execution, Visual knowledge boards, Community knowledge sharing, Document ingestion system, Digital learning platforms, Knowledge organization.

## 1. INTRODUCTION

Modern students today use a variety of digital tools for activities ranging from note taking to coding, document viewing and discussion forums. While these tools offer great functionality, they tend to work in isolation, which means that there is no integration of the applications or tools, and as such, students have to manage a plethora of tools, which leads to disjointed workflows and knowledge fragmentation. This fragmentation and switching between apps, greatly reduces the productivity of students.

Today, many students spend a large amount of time in digital environments creating content for their classes, communicating with instructors, and collaborating with their peers. Managing the multiple platforms, tools, and sources of information that make up their learning environments, however, can be frustrating and barriers to engagement. To help address this problem, this paper introduces ConnectEd, an integrated collaborative learning workspace that provides a single source for a core set of learning activities such as note taking, programming, idea organization, and community sharing. ConnectEd allows users to create hierarchical, outlinded notes that include executable code, idea maps and concept shelves organized on digital boards, and a social learning community for sharing knowledge. The ConnectEd workspace is intended to streamline digital learning environments, reduce the time required to find and switch between different applications, and aid in student knowledge management and understanding.

## 1.1 Problem Statement

Students today spend a lot of time using separate applications to manage the various activities involved in learning. These include tools for note taking, programming environments, viewers for PDF documents, discussion forums, and many other applications. As these applications are all separate, students struggle to get all the resources they need in one place. Relevant information is spread across many apps, so students struggle to revisit concepts they have learned some time ago and re-use content they have already created. Because apps are frequently different and have different interfaces, switching between them is a common source of distraction that breaks up learning flows.

## 1.2 Proposed Solution

ConnectEd solves this problem by bringing all collaborative learning activities and academic tools under one umbrella, a unified collaborative learning workspace. In ConnectEd, users can write notes with structures, write code inside their notes, and do mind mapping and sketching with knowledge boards. Users can also post questions in a community knowledge-sharing forum to ask for help, to share knowledge, to discuss topics with other

peers and experts. ConnectEd unifies all these features into one single platform, streamlining learning workflow, and helping learners to better organize their learning activities and assets, and thus facilitating more efficient learning and collaborative learning experiences.

## 2. SYSTEM ARCHITECTURE

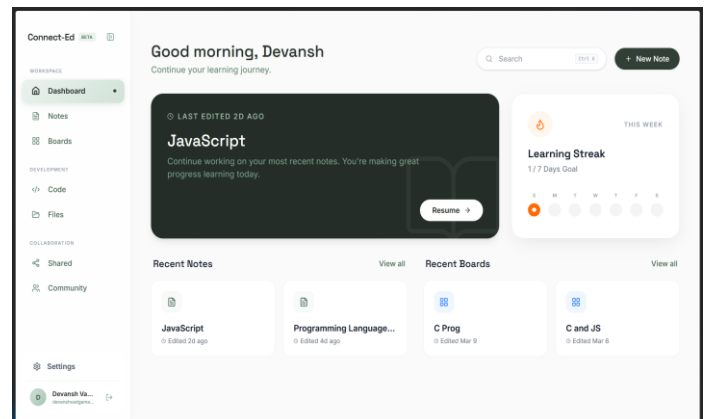
ConnectEd is designed as an integrated digital learning platform that combines multiple academic and development tools within a single workspace. The system architecture focuses on improving productivity, knowledge organization, and collaboration for students by reducing the need to switch between multiple applications. Instead of relying on separate tools for note-taking, coding practice, document viewing, and discussions, ConnectEd integrates these capabilities into a unified learning environment.

**Table -1:** ConnectEd System Modules

ConnectEd System Modules		
Module	Description	Purpose
Structured Notes System	Block-based notes supporting text and code	Organizes study materials
Code Execution Engine	Runs programming code within notes	Enables interactive learning
Knowledge Boards	Visual boards for connecting ideas	Helps conceptual understanding
Document Ingestion System	Processes PDFs and documents	Converts materials into notes
AI Processing Pipeline	Analyzes notes and code using AI models	Provides explanations, summaries, and debugging
Community Platform	Discussion, note sharing, code sharing	Supports collaborative learning

The core component of the system is the **structured notes module**, which allows users to create block-based notes containing text, code snippets, and references. This structure enables students to organize their learning materials efficiently while keeping related content together. The notes module acts as the primary workspace where users write, study, and interact with different types of information. In addition, the system includes an **embedded code execution environment** that allows users to run programming code directly within the notes

interface. This integration enables students to experiment with programming concepts while studying theoretical material, creating a more interactive learning experience.



**Fig. 1:** ConnectEd workspace interface

Another key component of the architecture is the **visual knowledge board module**, which provides a workspace where users can connect ideas and organize concepts visually. These boards help learners understand relationships between topics and build structured knowledge maps. ConnectEd also includes a **document ingestion system** that allows users to upload learning resources such as PDFs, presentations, and documents. These files are processed and converted into structured notes so that the content can be easily reviewed, organized, and reused during study sessions.

The system architecture also incorporates an **AI processing pipeline** that enhances the learning experience through intelligent content analysis. The pipeline is designed as a modular, multi-stage system that connects the user’s workspace with AI-powered language models, enabling a wide range of on-demand academic assistance features. The pipeline operates across four distinct stages: context collection, prompt construction, model inference, and response delivery.

In the **context collection stage**, when a user triggers an AI action (such as “Summarize”, “Explain”, or “Debug”), the platform automatically gathers relevant contextual data from the active workspace. This includes the content of the current note block, surrounding blocks within the same note, any code snippets present in the selection, metadata such as the programming language or subject tag, and document content if a file has been ingested into the session. By assembling rich, workspace-aware context rather than relying on isolated text, the pipeline ensures that AI responses are grounded in the student’s actual learning material.

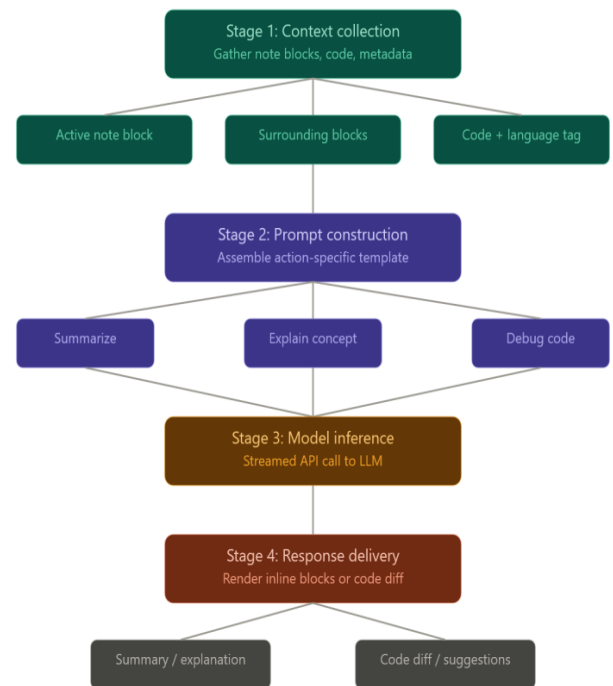
The second stage, **prompt construction**, involves assembling the collected context into a structured prompt that is sent to the underlying language model. The prompt template varies depending on the requested action. For a

summarization request, the prompt instructs the model to condense the provided note content into concise key points. For a concept explanation request, it asks the model to describe the highlighted term or passage in simple language suitable for a student learner. For a code analysis request, it directs the model to review the code, identify any logical or syntactic issues, and suggest improvements. This template-driven approach ensures that each type of AI action consistently produces relevant, task-appropriate output.

During the **model inference stage**, the constructed prompt is transmitted to a large language model via a secure API. The platform is designed to support integration with multiple AI model providers, allowing the system to be configured with different models depending on performance requirements and cost considerations. Responses are streamed back to the client as they are generated, minimizing perceived latency and enabling users to begin reading the output before the full response is complete. Error handling at this stage includes timeout detection, fallback messaging, and graceful degradation in the event that the AI service is temporarily unavailable.

In the final **response delivery stage**, the model's output is parsed and rendered within the ConnectEd interface. Plain text explanations and summaries are displayed as inline note blocks that can be edited, saved, or discarded by the user. Code suggestions are presented in a diff-style view, highlighting proposed changes so that learners can review and selectively apply them. This non-destructive delivery approach ensures that AI-generated content supplements the student's own work rather than overwriting it, keeping the learner in control of their notes at all times.

The AI pipeline supports three primary use cases within ConnectEd. The first is **note summarization**, where students can select a note or a section of content and request a condensed summary. This is particularly useful when reviewing large sets of study notes before an examination. The second use case is **concept explanation**, where students can highlight an unfamiliar term or passage and request a plain-language explanation, reducing the need to switch to an external search engine or reference tool. The third use case is **code analysis and debugging**, where students can submit a code block to the pipeline and receive feedback on correctness, efficiency, and style. This feature supports both beginners learning syntax and more experienced learners refining their programming skills. Together, these capabilities make the AI pipeline a central component of the ConnectEd learning experience, transforming passive note content into an interactive and intelligent study resource.



**Fig. 2:** AI Processing Pipeline - four-stage flow from context collection to response delivery

Finally, ConnectEd integrates a **community knowledge-sharing module** that enables users to ask questions, share notes, publish code snippets, and participate in discussions with other learners. This collaborative environment encourages knowledge exchange and allows students to learn from shared experiences and resources.

By integrating structured note-taking, code experimentation, AI-assisted learning, document processing, visual knowledge organization, and collaborative interaction, the overall architecture of ConnectEd creates a unified digital learning ecosystem that supports efficient and interactive education.

### 3. CONCLUSIONS

This paper presented **ConnectEd**, an integrated collaborative learning platform designed to simplify digital learning workflows for students. Modern learners often rely on multiple independent tools for note-taking, coding practice, document viewing, and online discussion, which can lead to fragmented workflows and scattered knowledge. ConnectEd addresses this issue by combining these essential academic tools into a single unified workspace.

The platform provides features such as **structured note-taking, embedded code execution, visual knowledge boards, document ingestion, and community-based knowledge sharing**. These components allow users to

create organized study materials, experiment with programming concepts, visually connect ideas, and collaborate with other learners within the same environment. In addition, the integration of an **AI processing pipeline** enhances the learning experience by enabling intelligent features such as content summarization, concept explanation, and code analysis.

By integrating knowledge creation, experimentation, visualization, and collaboration into one platform, ConnectEd provides a more efficient and connected learning ecosystem. The system helps students manage learning resources more effectively while encouraging interactive and collaborative study practices.

Future improvements to the platform may include advanced AI-based study assistance, improved collaboration features, and enhanced document processing capabilities to further support digital learning environments.

## ACKNOWLEDGEMENT

The authors would like to express their gratitude to the faculty members and mentors who provided guidance and support during the development of this project. Their valuable suggestions and encouragement helped in completing this research work successfully.

## REFERENCES

- [1] P. Brusilovsky and C. Peylo, "Adaptive and Intelligent Web-based Educational Systems," *International Journal of Artificial Intelligence in Education*, vol. 13, no. 2, pp. 159–172, 2003.
- [2] R. E. Mayer, *Multimedia Learning*, 2nd ed., Cambridge University Press, 2009.
- [3] T. Anderson, "Towards a Theory of Online Learning," in *Theory and Practice of Online Learning*, AU Press, 2008.
- [4] M. Alavi and D. E. Leidner, "Review: Knowledge Management and Knowledge Management Systems," *MIS Quarterly*, vol. 25, no. 1, pp. 107–136, 2001.