
OWLLM: A Learning Management System Prototype With LLM-assisted Course Content Interaction

Michael Aiello

An Honors Thesis Submitted to the Department of Computer Science and Honors College

Abstract: The increasing prevalence of Large Language Models (LLMs) presents both opportunities and challenges for education. This thesis explores the potential of LLMs to enhance course-based interaction by introducing OWLLM, a proof-of-concept web application that integrates a course-specific LLM chat interface alongside core Learning Management System (LMS) features. OWLLM was developed as a scaled-down LMS to provide a simple yet focused demonstration of LLM integration within educational workflows. Rather than embedding LLM capabilities into an existing LMS—which remains the long-term goal—this standalone prototype serves as a demonstration to inform future design, evaluation, and iteration of LLM-based educational tools. While the integration of AI into education is not new and has been evolving for some time, this thesis presents a contemporary view that distinguishes AI as an umbrella term encompassing LLMs, natural language processing (NLP), and their intersections with machine learning and deep learning. It addresses the ethical considerations and implementation challenges involved, emphasizing behavioral and intentional approaches toward ethically optimal applications. Key features of LLM technology make it especially well-suited for educational integration, supporting neurocognitive and psychological principles such as information processing, feedback loops, knowledge retention and consolidation, self-determination theory, and cognitive load theory. OWLLM leverages these principles through configurable course interaction commands and customizable Retrieval-Augmented Generation (RAG) system parameters. These RAG fundamentals are essential to OWLLM's ability to perform instruction-based prompt engineering that produces fine-tuned responses aligned with educational best practices, while mitigating common LLM misuse in learning environments. This thesis examines both the software engineering processes behind OWLLM's development and the ways strategically implemented LLM tools can enhance educational engagement and supplement traditional study practices.

Index Terms: Artificial Intelligence in Education (AIED), Large Language Models (LLM), Educational Software, Learning Management Systems (LMS), Blackboard, Scaled-Down LMS, Proof of Concept, AI Integration, Software Engineering, User Stories, CursorIDE, AI-Assisted Development, NodeJS, Express API, PostgreSQL, Supabase, Retrieval-Augmented Generation (RAG), MIME Types, Document Chunking, OpenAI, Vector Embeddings, Similarity Search, Command Triggers, Course Interaction Enhancement, Prompt Engineering, Response Fine-tuning, Natural Language Processing (NLP), Ethical AI, Neuro-cognitive Principles, Cognitive Load Theory, Self-Determination Theory, Knowledge Retention, Feedback Loops, Digital Equity, Accessibility, Costs, Scalability

1. Introduction

The rapid advancement and expanding accessibility of Large Language Model (LLM) technology have sparked considerable interest within the field of education regarding their potential integration. Faced with evolving educational landscapes marked by challenges such as increasing student-to-teacher ratios and concerns about student engagement, the exploration of innovative solutions has become paramount. Artificial intelligence (AI) technology, particularly LLMs that utilize natural language processing (NLP), offers promising avenues for impactful implementation in educational settings.

This project is motivated by the observed capabilities of increasingly powerful LLMs, which hold significant potential to enrich learning experiences. While various LLM applications are emerging, this work focuses on developing a tool directly aligned with course content and educational objectives. Key to this effort is the ability to configure commands and fine-tune responses to proactively address potential misuses of LLMs, including plagiarism and over-reliance.

This thesis introduces OWLLM, a proof-of-concept web application designed to simulate and supplement LMS functionality by embedding an LLM chat interface tightly linked to course materials. Rather than being integrated into an existing LMS, OWLLM includes its own simplified LMS components—such as course content delivery and user roles—to enable full control over the development environment and to support testing in a realistic context. This implementation was necessary to evaluate how a strategically deployed LLM could enhance course-based interactions and support independent study and homework practices.

In addition to a literature review covering foundational principles of AI in education, the primary focus of this thesis lies in the software engineering aspects of designing, developing, and deploying OWLLM. Ultimately, this work contributes to ongoing discussions about the role of LLMs in educational technology and how such tools may be thoughtfully implemented to support both learners and educators.

2. Literature Review

2.1. Introduction and Foundational Concepts

When the discussion of implementing AI tools in educational settings to improve learning outcomes is brought about, it is important to recognize that this is not a new concept and has been in practice for some time [1]. However, as these technologies evolve rapidly, their application in education continues to open new avenues to improve personalized learning, assessment, and interaction.

It is also crucial to distinguish between AI as a broad field and specific technologies, such as NLP and LLM, that operate under its umbrella. AI refers broadly to the simulation of human intelligence in machines programmed to think and learn, while NLP focuses on the interaction between computers and human language, and LLMs, such as OpenAI's GPT models, are deep learning models trained on large data sets to generate and interpret human-like language [2] [3].

Before engaging in a comparative review of literature, it is helpful to clarify foundational concepts. A common misconception is that AI represents machines with intelligence equivalent to that of humans. In reality, AI aims to replicate specific cognitive abilities—such as learning, problem-solving, and language comprehension—through algorithms and computational models [4]. Korteling et al. emphasize that while AI can simulate certain cognitive processes, it lacks the consciousness, empathy, and full contextual awareness of the human mind [5]. Nonetheless, mimicking elements of human intelligence remains a primary goal within AI research.

2.2. Machine Learning Background

In summary, although the incorporation of AI into education is not entirely new, the sophistication and accessibility of technologies such as NLP and LLMs have significantly expanded their potential. A clear understanding of these tools and their respective roles is essential for educators and policymakers aiming to implement them effectively and ethically.

The process by which a computer becomes “artificially intelligent” is primarily rooted in machine

learning (ML), a core subfield of AI that enables systems to improve performance on tasks through experience rather than explicit programming [6]. Unlike humans, who can adapt and learn through abstraction and intuition, computers must be given structured instructions through code. These instructions are formalized into algorithms—systematic sets of rules that allow machines to identify patterns within large volumes of data. Over time, these algorithms construct predictive models that adjust based on new input, enabling increasingly refined outputs through iterative learning cycles [7] [8]. Machine learning operates through various paradigms, such as supervised learning—where systems are trained on labeled data—and unsupervised learning—where they detect patterns in unlabeled data [9]. In all these methods, the accuracy of the resulting models depends heavily on both the quality of data and the presence of human feedback throughout training and evaluation phases [10].

As Alzubi et al. explain, algorithms serve as the foundational structures through which machines recognize trends, classify information, and make informed projections based on continuous exposure to datasets [6]. Importantly, while algorithms evolve with data, human oversight is necessary to assess the validity of outcomes, especially when applied to nuanced domains like education. This dependency underscores a hybrid relationship where AI complements but does not entirely replace human intelligence.

2.3. Role and Applications of NLP/LLMs in Education

In educational contexts, this distinction becomes especially relevant. NLP and LLMs have played central roles in enabling adaptive learning environments, automating assessments, and facilitating intelligent tutoring systems. For instance, Litman discusses the use of NLP in enhancing educational feedback mechanisms and evaluating classroom discourse [1]. Similarly, Wang et al. explore how AI-powered systems enable real-time, personalized learning trajectories for students, particularly in online and hybrid learning models [11].

Furthermore, Holmes et al. highlight how AI, when integrated thoughtfully, can complement human instruction rather than replace it—providing scalable tools for feedback, curriculum personalization, and learner engagement [12]. Zawacki-Richter et al. also conducted a systematic review revealing that NLP-driven applications, such as chatbots, virtual tutors, and writing assistants, are among the most impactful AI tools used in higher education today [13].

In the context of language learning, LLMs show immense promise. Their ability to understand context, offer immediate feedback, and generate naturalistic dialogue significantly enhances language acquisition. Jurafsky and Martin outline how transformer-based models such as BERT and GPT-3 have revolutionized tasks like grammar correction, text classification, and semantic analysis [3].

When connecting machine learning and AI to the educational context, it becomes essential to ask why such tools are needed. The overarching intent of integrating AI in education is to “supplement, support, and nurture” student learning experiences by tailoring educational content and feedback to individual learners’ needs. Traditional educational metrics—such as grades and test scores—offer limited insight into the process of learning. In contrast, machine learning systems can utilize rich linguistic and behavioral data from student writing and communication to make inferences about understanding, misconceptions, and progression [14]. This is where NLP plays a crucial role.

NLP enables educational systems to parse, interpret, and generate human language, making it a powerful mechanism for interpreting student-generated texts such as essays, discussion responses, or written assessments. Through NLP, systems can detect structural, syntactic, and semantic features of student writing, allowing for scalable, formative feedback and more nuanced evaluation [15] [16]. As transformer-based models like GPT and BERT become increasingly sophisticated, their ability to analyze discourse, offer suggestions, and even scaffold learning through dialogue-based interactions has shown promising applications in intelligent tutoring systems [17].

Now that foundational topics related to AI, NLP, and ML have been introduced and contextualized in education, we can begin to explore the current academic discourse in more depth. This phase

of the literature review aims to build a rationale for the proposed research by demonstrating its significance within the context of existing studies. Through synthesizing a range of sources, the goal is to evaluate key themes, recognize gaps or divergences in the literature, and lay the groundwork for a research framework by highlighting major ideas and their interconnections.

A critical contribution to this conversation is the work by Khensous and Labeled, titled *Exploring the Evolution and Applications of Natural Language Processing in Education* [18]. The authors present NLP as a multidisciplinary field combining elements of computer science, AI, information engineering, and linguistics. Within the educational context, they identify three significant areas of application: Recommender Systems (RS), Sentiment Analysis (SA), and Chatbots. Each of these tools supports different pedagogical needs and demonstrates the increasing versatility of NLP technologies in academic settings.

In the realm of Recommender Systems, the goal is to enhance educational outcomes by guiding learners toward appropriate resources and strategies. RSs function by recommending materials such as articles and books, predicting academic performance, and identifying learning deficiencies that can be proactively addressed. Despite their promise, the authors point out a lack of consensus on the most effective algorithms for educational recommendation tasks and highlight the need for further research and optimization of these models [18] [19]. This aligns with findings by Kumar and Thakur, who emphasize that most RSs in education are adapted from commercial systems, often without sufficient calibration for pedagogical effectiveness [20].

Chatbots represent another widely discussed application of NLP in education. As conversational agents, they simulate human dialogue to provide real-time interaction with students. Their strength lies in scalability—offering support to large numbers of students without requiring direct instructor involvement. However, the limitations are also significant: chatbots may struggle to interpret complex, open-ended queries or to respond with empathy and nuanced understanding [21]. The article by Khensous et al. further acknowledges public concern over the potential for these technologies to replace educators. It clarifies, however, that the purpose of chatbots is to augment instructional capacities—not to substitute them. By automating low-level cognitive tasks such as FAQ answering or basic guidance, chatbots can reduce teacher workload, thereby allowing instructors to focus on higher-order pedagogical engagement [18].

In parallel, chatbots also show potential as intelligent tutoring assistants, capable of offering personalized feedback, guiding learners at their own pace, and even fostering greater independence among students who prefer self-regulated learning environments [22]. This reflects broader movements in AI-enhanced learning, where customization and autonomy are increasingly prioritized.

2.4. Fundamental and Advanced NLP Techniques

In everyday human conversation and written communication, people effortlessly perform a wide range of linguistic tasks without conscious thought—deciphering grammar, understanding context, and interpreting meaning happen almost instantaneously, unless the material is particularly complex or ambiguous. NLP strives to replicate these cognitive feats within machines by developing algorithms capable of processing and interpreting human language in a structured and meaningful way [23]. NLP bridges the gap between natural human communication and machine understanding by enabling computers to analyze, comprehend, and, in some cases, generate human language in real-time.

To understand how NLP operates, one can draw parallels between human linguistic competence—such as understanding parts of speech or contextual cues—and the layered steps that machines must undertake to achieve similar comprehension. A fundamental task in NLP is tokenization, where a body of text is broken down into smaller units like words or phrases. These tokens become the base units for further analysis. Tokenization is typically followed by stop word removal, which filters out common words such as "the," "is," or "and" that contribute little to the semantic meaning of the text [23] [24].

Next, stemming and lemmatization are employed to reduce words to their base or root form. For

example, "running," "ran," and "runs" can all be reduced to "run." This standardization is critical in helping algorithms recognize different forms of the same concept. Part-of-speech (POS) tagging assigns labels to each token to indicate whether it is a noun, verb, adjective, etc., and Named Entity Recognition (NER) identifies specific categories of key information such as person names, locations, organizations, and dates [25]. These techniques work collectively to provide the machine with a linguistically annotated version of the input data, much like how humans intuitively grasp grammar and context.

Additionally, statistical methods like Bag-of-Words (BoW) and TF-IDF (Term Frequency–Inverse Document Frequency) are often used in preprocessing stages to represent textual data numerically. The BoW approach treats text as a multiset of its words, often disregarding grammar and word order but capturing frequency [26]. In contrast, TF-IDF not only considers word frequency within a document but also accounts for how unique or informative a word is relative to a corpus. This gives more weight to rare but significant words, making it a powerful tool in tasks like document classification or keyword extraction [27].

Once the text is preprocessed and structured in this manner, the system can then move on to more advanced and interpretive layers of NLP. These include syntactic analysis, which focuses on sentence structure and grammatical relationships; sentiment analysis, which determines the emotional tone behind a body of text; and semantic analysis, which seeks to understand the deeper meaning and intent of words and phrases within their context [28]. These stages are particularly vital in educational applications of NLP and LLMs, as they allow for the development of nuanced, personalized feedback systems.

2.5. Sentiment Analysis and Learner Emotion Detection

By gauging both the cognitive and emotional dimensions of student responses, AI-driven tools can tailor educational content, suggestions, and guidance to the learner's current state of understanding and motivation. For example, if a student expresses confusion or frustration in written assignments or forum posts, a sentiment analysis module could detect this and prompt the system to respond with clarification or encouragement. Similarly, semantic understanding ensures that student misconceptions are not only identified but also addressed in ways that align with how the student is currently processing the material [29].

As a simple definition, Sentiment Analysis (SA) is a form of NLP that interprets the emotional tone of written or spoken language. In education, SA can be leveraged to detect signs of learner dissatisfaction, disengagement, or risk of early dropout [18]. Through regular analysis of student submissions, forum posts, or even voice inputs, sentiment-aware systems can alert educators to shifts in motivation or attitude. This proactive monitoring can facilitate timely interventions, a point also reinforced by recent studies emphasizing the role of emotional analytics in improving learner retention and satisfaction [30] [31].

In discussion of SA, to introduce a significant addition to the ongoing discourse around NLP-based AI in education, there is a book 'Advancing Natural Language Processing in Educational Assessment', edited by Victoria Yaneva and Mathias von Davier [34]. This book will be referenced later for its significance; however, it should be noted that it does notably lack substantial discussion of sentiment analysis, which is surprising given its relevance to educational assessment. Emotional and affective factors such as motivation, anxiety, and frustration play a substantial role in learning outcomes, and sentiment analysis has the potential to capture and respond to these dimensions in real time. Several other studies argue for the inclusion of affective analytics as an essential component of AI-enhanced education, noting that students' emotional states significantly impact engagement and comprehension [32] [33].

A critical insight from the book that aligns closely with this concern is its acknowledgment that the predictive accuracy of NLP-based indices depends heavily on the level of student engagement. This highlights a major consideration for future research: the effectiveness of NLP systems is not purely a matter of linguistic accuracy or technical performance, but also of contextual responsiveness—how well the system adapts to the learner's cognitive and emotional state [34][35].

This points to the need for integrative models that combine syntactic and semantic analysis with affective computing approaches to holistically support student development.

To build upon the previously discussed concept of learner engagement and its influence on adaptive educational technology, the conversation now shifts to a key dimension of NLP in education: sentiment analysis and learner emotion detection. These concepts are critically examined in the article *Two Decades of Artificial Intelligence in Education: Contributors, Collaborations, Research Topics, Challenges, and Future Directions* by Chen, Zou, Xie, Cheng, and Liu [36]. Although this article does not focus exclusively on NLP in the way earlier sources do, it nonetheless brings renewed attention to recommender systems and provides a more in-depth discussion of sentiment analysis within the broader scope of AIED.

In reference to the earlier topic—using NLP to address issues like early student dropouts—the authors emphasize that AI-powered educational systems can be instrumental in monitoring classroom dynamics and assessing student engagement in real-time. This, in turn, enables educators and institutions to identify at-risk students early and intervene before learning disruption escalates [36]. A particularly novel contribution of the article is its meta-research approach, in which the authors employed machine learning tools to analyze the past two decades of AIED literature. This “research on research” revealed that a persistent theme in the AIED field is the study of learner affect and emotional diversity across varied learning scenarios.

Findings indicate that emotions such as frustration, confusion, motivation, and anxiety frequently appear in educational research as core determinants of learning outcomes. Crucially, the authors noted an evolving trend wherein Intelligent Tutoring Systems (ITS) began integrating NLP-assisted tools specifically designed to manage emotional variability in learners. This finding reinforces insights from von Davier and Yaneva [37], wherein adaptive NLP systems such as iStart were shown to facilitate custom feedback loops aimed at enhancing reading comprehension. In this context, NLP’s role in providing automatic, real-time feedback becomes a central benefit—not only for academic improvement but also for emotional regulation and motivation [37] [38] [39].

An illustrative example mentioned in Chen et al.’s article is the Genie Tutor, a language-learning system that delivers immediate feedback on language errors. Much like the stealth literacy assessments in iStart, Genie Tutor creates a personalized, responsive learning environment that can help learners manage anxiety and build confidence. These systems represent a feedback-driven model of learning where adaptive interactions can dynamically accommodate both the cognitive and affective needs of the student [38] [40].

While these technologies do not definitively prevent disengagement or dropout, they serve as proactive tools to mitigate emotional barriers to learning. This is particularly valuable in digital learning environments where instructors may not be physically present to detect signs of disinterest or distress. The article concludes by proposing an innovative direction for future research: combining sensor-based emotion detection (e.g., facial recognition, physiological sensors) with textual sentiment analysis via NLP. Such a dual-layered emotion detection framework could offer more accurate insights into learner experience and be used to fine-tune AI-based educational interventions [38] [41].

In essence, the article underscores a critical but underexplored insight: while NLP excels at processing text and offering syntactic and semantic feedback, it also holds untapped potential as a tool for emotional diagnostics and support. The correlation between learner emotion and NLP-based systems is not yet fully understood, but emerging evidence suggests that these tools could provide scalable, personalized support structures capable of addressing widespread issues like learner dissatisfaction, disengagement, and dropouts—problems that are increasingly urgent in the modern, digitized educational landscape [42].

2.6. Case Study: The iStart Platform and Adaptive/Stealth Assessment

Although previously mentioned to be lacking in one aspect of NLP, “Advancing Natural Language Processing in Educational Assessment”, edited by Victoria Yaneva and Mathias von Davier [34] is still one of the most comprehensive efforts to date in documenting and analyzing real-world

implementations of NLP technologies in educational settings. The collection of chapters provides critical insights across several dimensions, including the historical evolution of NLP in assessment, issues of validity and fairness, the integration of emerging technologies, and the implications for personalization and feedback in educational environments.

One of the standout chapters in this volume is the discussion of Stealth Literacy Assessment, which not only deepens the reader's conceptual understanding of NLP but also introduces a highly sophisticated use case: the iStart platform, an intelligent tutoring system built to support active reading and comprehension strategies. iStart, short for Interactive Strategy Training for Active Reading and Thinking, blends instructional video lessons, coached practice, and game-based interactivity to foster student engagement and skill development [43] [44]. What sets this system apart is its dual-layered feedback approach, coined in the book as inner-loop and outer-loop adaptivity.

Inner-loop feedback involves immediate, real-time responses to student actions within a task, similar to formative assessment techniques used in classrooms. In contrast, outer-loop adaptivity is more strategic, using the student's performance data to determine the sequence and difficulty of future learning tasks. When these two feedback mechanisms are used in tandem, the system can adapt fluidly to each learner's needs, optimizing the instructional pathway without interrupting the learning experience. This distinction aligns with principles found in intelligent tutoring system design literature, which has consistently highlighted the effectiveness of adaptive feedback in promoting deeper learning and long-term retention [45] [46].

The stealth aspect of this literacy assessment refers to its ability to evaluate student skills continuously without overtly labeling or interrupting the learning process. This unobtrusive assessment model allows for constant measurement of progress while preserving the natural flow of engagement—a balance that traditional testing methods struggle to maintain. It is precisely this integration of assessment within instruction that makes NLP-driven platforms like iStart both scalable and effective for diverse educational contexts [47]. By embedding diagnostic mechanisms into the learning content itself, such systems can support personalized learning trajectories that reflect each student's evolving needs.

Furthermore, this model illustrates a broader shift in educational assessment from summative evaluation to formative, adaptive learning environments. Such platforms do not merely provide content delivery—they act as responsive learning companions. As AI technologies continue to evolve, this adaptive capacity is expected to become more sophisticated, particularly as systems gain access to more diverse and multimodal data inputs such as speech, facial expressions, and behavioral patterns [48]. In sum, the iStart case demonstrates the tangible benefits of applying NLP to create intelligent feedback loops that drive both instruction and assessment forward simultaneously.

2.7. Scalability, Methodological Challenges, and Literature Comparison

Taking a broader view of the implications of the iStart system and its use of NLP-based adaptive feedback, the feasibility of implementing stealth assessments on a wider scale becomes an increasingly prominent topic. The success of platforms like iStart highlights how unobtrusive, continuous assessment methods can be embedded into instructional content to better align with individualized learning needs. However, while early findings from studies on iStart suggest that such assessments—particularly those focused on students' language use and comprehension strategies—are relatively reliable, there remain limitations in generalizability. As noted in "Advancing Natural Language Processing in Educational Assessment", despite the robust structure of the system, results across different demographics, educational settings, and curricular domains are still too generalized to serve as conclusive evidence for universal scalability [49]. The authors argue that additional studies are essential to better understand how the predictive accuracy of NLP indices might fluctuate depending on variables like language background, learning preferences, and socioeconomic factors [37].

This opens a critical methodological dialogue about how stealth assessments should be im-

plemented. The core challenge lies in designing NLP systems that are both flexible and robust—capable of adjusting to nuanced learner differences while maintaining consistency and validity in assessment results. As suggested by recent work in the field, adaptive systems must be trained on diversified datasets and rigorously validated across varied learning contexts in order to prevent biased conclusions and ensure ethical implementation in real-world educational settings [51].

A comparison between the article by Khensous et al. and the book by von Davier and Yaneva [43] brings several meaningful contrasts to light. The article is more speculative and exploratory, offering potential use cases for NLP applications in education, such as chatbots, recommender systems, and sentiment analysis. These tools are discussed with theoretical justification but without detailed empirical support. In contrast, the book takes a more grounded approach, providing rich case studies and evidence-based analyses of existing platforms like iStart. It dives into the mechanics of how NLP technologies can enhance student learning, with a particular focus on embedded assessments and interactive learning systems.

2.8. Ethical Concerns and Misuse of AI

However, the increasing integration of AI in educational environments also raises ethical and pedagogical questions. Concerns about data privacy, algorithmic bias, and over-reliance on machine-generated outputs are widely discussed in current scholarship. Binns et al. argue that transparency and accountability in AI systems must be prioritized, especially in contexts affecting student evaluation and progression [52]. Likewise, Selwyn warns against uncritical adoption of AI, urging institutions to remain aware of the broader societal implications and power dynamics at play [53].

Within the growing discourse on dropout rates in education—often attributed to a combination of disengagement and complex emotional or cognitive challenges—an emerging concern is the misuse of NLP technologies by students. Increasingly, learners confronted with academic difficulties are turning to AI-powered tools not simply for assistance, but for circumventing academic responsibilities. This growing reliance on NLP technologies for dishonest purposes, such as auto-generating entire essays or answers, undermines the principles of academic integrity and may exacerbate pre-existing motivational issues. As these tools become deeply embedded in educational ecosystems, the ethical implications of their use cannot be ignored. This calls for a proactive framework to detect, mitigate, and manage academic dishonesty while continuing to harness NLP's educational potential [54] [55].

2.9. Large Language Models (LLMs) like ChatGPT: Opportunities, Challenges, and Frameworks

A key contribution to this conversation is the article by Kasneci et al., titled "ChatGPT for good?: On opportunities and challenges of large language models for education", which explores both the promise and the perils of LLMs such as ChatGPT [56]. While earlier sources have discussed AI chatbots in education, this article stands out due to its timely focus on ChatGPT's mainstream adoption and its multifaceted role in academic contexts. As the authors detail, ChatGPT is capable of performing a variety of tasks relevant to education—ranging from essay writing, tutoring, summarization, to code generation, language translation, and content synthesis. These applications offer significant benefits in terms of scalability and accessibility for students across disciplines.

However, the article also articulates serious ethical, pedagogical, and technical concerns—among them, risks of bias, misinformation, plagiarism, and erosion of critical thinking. These risks are especially relevant to the broader concerns of this literature review, as they pertain not just to academic dishonesty but also to the depersonalization of the learning process. The tension lies in the fact that while AI-generated content can be efficient and comprehensive, it may also discourage deeper cognitive engagement and hinder the development of essential academic skills [56] [57].

In an effort to address these challenges, Kasneci et al. propose a framework of human-AI collaboration—emphasizing the importance of standardization, transparency, and reflective practice. Among their recommendations is the creation of audit trails that document how students

interact with ChatGPT, including queries submitted and how the content was refined, interpreted, or questioned. Students should also be encouraged to submit reflection reports that critically assess their use of AI assistance and identify sections where human judgment, creativity, or contextual understanding was required. This hybrid model offers a compromise between AI utility and academic integrity, where the student is held accountable while also benefiting from the tool's affordances [56].

The article also recommends structured trial programs, where students are explicitly taught how to use AI tools ethically and responsibly. This not only enhances awareness of AI's limitations—such as its tendency to generate hallucinated or unreferenced claims—but also promotes meta-cognitive awareness of the learning process itself [58]. Ultimately, the article underscores that while ChatGPT and similar tools pose clear risks, they can also become powerful allies in education when guided by responsible use frameworks that blend machine efficiency with human originality .

This discussion ties back to the foundational goals of NLP in education—to support, supplement, and enhance student learning. As such, the pathway forward may not be to restrict access to AI technologies, but to design robust pedagogical ecosystems where AI complements rather than replaces student agency, creativity, and ethical responsibility [59].

To further build on the conversation surrounding ChatGPT's role in education, Rane's article, titled "Enhancing the Quality of Teaching and Learning through ChatGPT and Similar Large Language Models: Challenges, Future Prospects, and Ethical Considerations in Education", provides a nuanced exploration of the potential and challenges of these AI tools in educational contexts [60] . While echoing many of the ideas presented earlier, Rane's work emphasizes the promising applications of ChatGPT and similar technologies. For instance, AI is highlighted as a tool with the capacity to customize assignments, tailor learning experiences to the individual needs of students, and analyze performance data to identify areas of struggle [60] . These capabilities position AI as a powerful asset for educators, allowing them to offer personalized feedback and monitor student progress over time.

A particularly important aspect of Rane's analysis is the integration of personalized learning and augmented feedback, which are consistent themes across previous sources like the iStart system and ChatGPT. The article underscores the vital role of immediate feedback, which helps students recognize errors, refine their understanding, and foster deeper learning. Rane also notes that AI's ability to track students' progress enables educators to better pinpoint areas where additional support is needed. This adaptive learning framework aligns with findings from other studies that emphasize real-time, responsive feedback as a cornerstone of successful educational outcomes [60] . Furthermore, Rane brings attention to the collaborative role of humans and AI, reinforcing that human educators remain indispensable for providing emotional support, guidance, and mentorship—elements that NLP technologies cannot replicate. This dual approach, blending AI capabilities with human intervention, is consistently championed throughout the literature as the ideal model for educational enhancement.

2.10. Conclusion of Literature Review

In conclusion, while this literature review does not purport to cover every aspect of the intersection between AI, NLP, and education, it effectively sets the stage for understanding the key connections and overarching ideas that pervade current research. As outlined, the core themes across the reviewed literature include personalized learning experiences, adaptive feedback mechanisms, and the use of NLP to assess and respond to academic and emotional needs. These advances, however, must be tempered by an understanding of the ethical implications and the continued necessity of human involvement in the educational process. In synthesizing these sources, a conceptual framework emerges that highlights the promise of AI technologies in reshaping education, while simultaneously acknowledging the complex challenges associated with their implementation [61].

3. Research Methodology - OWLLM Conceptual Foundations

3.1. Rationale

The rationale for developing and integrating a supplemental LLM tool into an LMS is grounded in insights from the previous literature review. To specifically address the software development aspects of this project, there are two key benefits of LLM technology that can be properly taken advantage of for education. First, an LLM chat interface can be placed in close proximity to course material. Second, an LLM can manage and load contextualized data from course material or custom input, enabling it to fine-tune responses. This fine-tuning capability, in addition to being in close proximity to course material, can be fully utilized properly in aiming at avoiding the main common misuses of the technology in education. This potential of LLMs to simplify course interactions by being implemented in close proximity to course materials and by providing fine-tuned responses based on context from the course materials is to be highlighted as the main design principle and driving force behind the applications development.

3.2. Web Application Technology

Web applications represent a fundamental shift in how software is delivered and accessed, making them an ideal choice for educational technology projects like OWLLM. Unlike traditional desktop applications that require installation and updates on individual machines, web applications run entirely within web browsers, providing universal accessibility across different devices and operating systems. This accessibility is particularly crucial in educational settings where students and educators may use a wide variety of devices and platforms.

The web application approach offers several key advantages for the OWLLM project. First, it eliminates the need for complex installation procedures, allowing users to access the system immediately through any modern web browser. This reduces technical barriers and ensures that the tool can be easily adopted without disrupting existing educational workflows. Second, web applications support real-time updates and improvements without requiring users to download or install new versions, ensuring all users benefit from the latest features and refinements simultaneously.

Moreover, web applications are well-suited to integrate with existing Learning Management Systems (LMS) such as Blackboard or Moodle. For proprietary platforms like Blackboard, integration is typically enabled via Learning Tools Interoperability (LTI) standards and REST APIs. While it is technically feasible to integrate OWLLM with Blackboard as an external LTI-compliant tool, deeper customization is restricted by Blackboard's proprietary nature and the degree of control allowed by the hosting institution. Access to Blackboard's APIs often requires administrative approval, and full plugin development is not supported outside of Anthology's internal ecosystem.

In contrast, open-source LMS platforms such as Moodle offer a more extensible environment. Moodle's architecture allows for custom plugin development and deeper integration with internal systems. This makes Moodle a more flexible platform for testing and embedding tools like OWLLM within an LMS interface, providing a greater degree of control over user interaction and course-specific customization.

Given these considerations, OWLLM's web-based architecture was chosen not only for its universal accessibility and ease of deployment, but also for its compatibility with modern LMS integration standards. Whether deployed as a standalone educational assistant or integrated into LMS platforms through LTI protocols, the web application model provides a scalable and platform-agnostic foundation for OWLLM's educational use case.

3.3. Large Language Model Fundamentals and RAG System

Large Language Models (LLMs) represent a significant advancement in artificial intelligence, capable of understanding and generating human-like text based on patterns learned from vast amounts of training data. These models operate by processing text through multiple layers of neural networks, where each layer extracts increasingly complex patterns and relationships within

the language. The fundamental unit of processing in LLMs is the "token," which represents a piece of text that can be as small as a single character or as large as a complete word, depending on the language and context.

Tokens serve as the building blocks that LLMs use to understand and generate text. When a user inputs a question or statement, the LLM breaks this text into tokens, processes them through its neural network layers, and generates a response token by token. The quality and relevance of these responses depend heavily on the model's training data and the specific context provided to it. However, LLMs have limitations when it comes to accessing current or domain-specific information, as their knowledge is limited to their training data cutoff date.

To address these limitations and enhance the educational utility of LLMs, this project implements a Retrieval Augmented Generation (RAG) system. RAG represents a sophisticated approach that combines the generative capabilities of LLMs with external knowledge retrieval mechanisms. The RAG system operates through several key components working in concert to provide contextually relevant and accurate responses.

The first component involves document processing and chunking. When course materials are uploaded to the system, they are automatically divided into smaller, manageable text segments called "chunks." These chunks typically contain 1000 characters with a 200-character overlap between consecutive chunks, ensuring that important concepts aren't split across boundaries while maintaining context continuity. This chunking strategy is crucial for maintaining the semantic integrity of the original content while making it searchable and retrievable.

The second component is vector embedding generation. Each text chunk is converted into a mathematical representation called a vector embedding using specialized embedding models. These embeddings capture the semantic meaning of the text in a high-dimensional space (typically 1536 dimensions for this project), where similar concepts are positioned closer together. This mathematical representation enables the system to perform semantic searches, finding content that is conceptually related even if it doesn't contain the exact same words.

The third component is the retrieval mechanism. When a user asks a question, the system generates an embedding for the query and searches through the stored document embeddings to find the most relevant chunks. This search uses cosine similarity, a mathematical measure that determines how closely related two vectors are in the high-dimensional space. The system retrieves the top-k most similar chunks (typically 5 chunks) that exceed a minimum similarity threshold (default 0.15), ensuring that only highly relevant content is considered for response generation.

The final component is response generation. The retrieved relevant chunks are combined with the user's original question and sent to the LLM as context. The LLM then generates a response that incorporates information from the retrieved documents while maintaining natural language flow. This approach ensures that responses are not only accurate and up-to-date but also grounded in the specific course materials relevant to the user's query.

The RAG system also incorporates several configurable parameters that allow for fine-tuning based on specific educational contexts. The temperature parameter (set to 0.7 for this project) controls the creativity and randomness of responses, with higher values producing more creative but potentially less focused responses. The maximum token limit (2000 tokens) ensures responses remain concise and manageable while providing sufficient detail. Similarity thresholds and top-k settings can be adjusted to balance between response relevance and comprehensiveness.

3.4. Technical Evaluation and Testing Methodology

The evaluation approach for the OWLLM project focuses on assessing how well the system meets its intended educational objectives rather than traditional software testing metrics. Given the constraints of a thesis project timeline, the evaluation methodology emphasizes qualitative assessment of usability and effectiveness in meeting specific use cases rather than conducting large-scale longitudinal studies with human subjects.

The evaluation process encompasses several key areas of assessment. First, the system's

ability to handle course content upload and processing is evaluated through testing various document formats and sizes. This includes verifying that the RAG system can effectively process and embed different types of educational materials, from text documents to multimedia content, ensuring that the vector embeddings accurately represent the semantic content of the uploaded materials.

Second, the evaluation examines user interactions with course content through the chat interface. This involves testing the system's ability to understand and respond to various types of educational queries, from simple factual questions to complex conceptual explanations. The assessment focuses on response relevance, accuracy, and helpfulness in educational contexts, ensuring that the AI provides meaningful assistance to both students and educators.

Third, the evaluation includes testing of command triggers and specialized features. This involves verifying that the system correctly identifies and responds to specific commands, such as relevance analysis requests or document-specific searches. The assessment ensures that these advanced features function as intended and provide valuable educational insights.

Fourth, the evaluation methodology includes comprehensive testing of the system's ability to satisfy the objectives and use cases outlined in the subsequent section. This involves systematically testing each identified use case to determine whether the system can effectively address the specific needs and scenarios described. The evaluation provides qualitative confirmation of whether the proof-of-concept tool can or cannot satisfy these user stories, serving as the primary metric for project success.

To support this evaluation process, the application incorporates a built-in feedback system that allows users to provide qualitative assessments of AI responses. This feedback mechanism enables the collection of user perspectives on response quality, relevance, and helpfulness, providing valuable insights into how well the tool meets user needs and expectations across different educational contexts. The feedback system serves as a key component of the evaluation methodology, offering real-world data on system performance and user satisfaction.

The evaluation approach recognizes that the success of an educational technology tool cannot be measured solely through technical metrics, but must also consider the qualitative aspects of user experience and educational effectiveness. By focusing on usability assessment and objective satisfaction rather than traditional software testing, the evaluation methodology aligns with the project's goal of creating a practical, user-centered educational tool.

3.5. Behavioral Framework for LLM-Assisted Learning

The OWLLM system is conceptualized and engineered not merely as a technical solution, but as a prototype thoughtfully grounded in established psychological and neurocognitive theories of learning. Its core aim is to enhance student engagement, refine study practices, and mitigate the potential for misuse of Large Language Models (LLMs) in educational contexts. This section elaborates on how OWLLM's design principles and features are intentionally aligned with Self-Determination Theory (SDT) and Cognitive Load Theory (CLT), alongside broader neurocognitive principles, to cultivate more effective and intrinsically motivated learning experiences.

3.5.1. Enhancing Motivation through Self-Determination Theory (SDT)

Self-Determination Theory (SDT), a prominent framework in motivational psychology, posits that intrinsic motivation—which is vital for sustained engagement and deep learning—is fostered by satisfying three fundamental psychological needs: autonomy, competence, and relatedness [62]. OWLLM's design actively addresses these needs to create a more compelling and effective learning environment.

Regarding autonomy, OWLLM empowers students by enabling them to self-direct their learning paths. Students can freely navigate through course materials, pose questions, and, crucially, save their personalized notes and specific questions using intuitive commands like `\note` or `\question`. This functionality allows learners to manage their own learning artifacts, thereby fostering a profound sense of ownership over their educational journey. Furthermore, the system's

encouragement of reflection reports on AI usage supports metacognitive awareness and self-directed learning, aligning with the idea that students should critically assess their use of AI assistance and identify areas requiring human judgment.

To cultivate a sense of competence, OWLLM provides personalized and adaptive feedback. The Retrieval Augmented Generation (RAG) system is central to this, delivering contextually relevant and accurate responses directly derived from the specific course materials. This adaptive scaffolding, reminiscent of operating within Vygotsky's Zone of Proximal Development (ZPD) [63, 64], aids students in tackling complex concepts by simplifying explanations or breaking them into digestible steps. The system also transparently indicates the source of its responses—whether from course material, general knowledge, or a combination—which builds trust and clarifies the scope of the AI's knowledge base. This transparency empowers students to refine their understanding and recognize any potential errors, aligning with the brain's error detection and reinforcement learning mechanisms.

While fostering direct human-AI social relatedness is a nuanced area, OWLLM simulates natural dialogue through its chat interface, making interactions feel more personalized and engaging. A key feature promoting relatedness and motivation is the `\relevance` command, which connects course content to students' academic and career interests. By demonstrating the practical value and personal relevance of the material, this command serves as a powerful motivational tool, designed to pique a user's drive, passion, or curiosity related to their goals. Future iterations could further enhance this by integrating collaborative learning features such as group chat spaces or shared note-taking environments, thereby nurturing a sense of connection within the broader learning community.

3.5.2. Optimizing Learning Efficiency with Cognitive Load Theory (CLT)

Cognitive Load Theory (CLT), introduced by Sweller [65], provides a framework for understanding how the limitations of working memory impact learning efficiency. CLT categorizes cognitive load into three types: intrinsic, extraneous, and germane. OWLLM's design specifically aims to optimize these loads to facilitate more effective learning.

Reducing extraneous cognitive load, which typically arises from distractions or poorly designed interfaces, is a key objective. OWLLM achieves this through its intuitive and clean user interface (UI), which prioritizes simplicity and accessibility. By placing the LLM chat interface in close proximity to course material, the system eliminates the need for students to switch between multiple platforms or contexts, significantly minimizing perceptual distractions and cognitive switching costs. The structured nature of responses from the LLM also reduces the mental effort required to process information, contributing to a smoother learning experience.

Managing intrinsic cognitive load, which relates to the inherent complexity of the subject matter, is addressed through OWLLM's adaptive capabilities. The RAG system breaks down complex course documents into smaller, manageable "chunks", allowing the LLM to provide adaptive scaffolding by simplifying explanations or breaking down concepts into digestible steps. The ability to fine-tune the LLM's responses further enables tailoring the complexity to suit the specific educational context and learner's proficiency level.

Finally, optimizing germane cognitive load—the mental effort directed toward schema formation and deeper learning—is encouraged by OWLLM through features that promote metacognition and active problem-solving strategies. The ability for students to save `\notes` and `\questions` facilitates active processing and engagement with the material. The `\relevance` command encourages students to think about the broader implications and connections of course content to their personal interests, fostering deeper cognitive engagement. Furthermore, future enhancements could incorporate Socratic questioning techniques [66, 67] or adaptive retrieval prompts, such as AI-generated quizzes, to actively stimulate reflection and understanding, thereby maximizing the mental effort dedicated to schema development and long-term retention.

3.5.3. Alignment with Neurocognitive Processes for Enhanced Learning

The fundamental design of OWLLM also considers how LLM-assisted learning aligns with and leverages the brain's inherent functions for sensory processing, attention, memory, and higher-order reasoning. This alignment aims to create a learning environment that is not only efficient but also neurocognitively supportive.

In terms of sensory processing and attention, the clean and intuitive UI design, along with the clear presentation of content within OWLLM, aims to enhance visual processing efficiency. This directly supports initial retinal processing, visual encoding, and the dorsal and ventral processing streams responsible for recognizing "where" elements are on the screen and "what" they represent [68, 69]. By minimizing extraneous distractions, the system aids the prefrontal-parietal networks in maintaining selective attention and focused engagement with the AI-generated content [70, 71].

For working memory and comprehension, OWLLM's structured content delivery, especially through the RAG system's chunking strategy, is designed to present information in manageable segments, thereby enhancing encoding efficiency within the working memory system. The LLM's capacity to generate logically structured and semantically rich responses supports the linguistic parsing functions of key brain regions like Broca's and Wernicke's areas [72, 73]. Broca's area engages in syntactic parsing and logical structuring [74, 75], while Wernicke's area facilitates lexical retrieval and semantic categorization [76, 77], both crucial for deep comprehension. The potential for future multimodal content support could further engage the angular gyrus for conceptual mapping and integrating information across different sensory modalities [78, 79].

Regarding memory encoding and retrieval, OWLLM facilitates elaborative encoding by allowing students to save notes and questions, which encourages linking new information to their existing knowledge base [80]. The context-aware responses from the LLM, firmly grounded in course materials, support the consolidation of new semantic knowledge into long-term memory, a process primarily mediated by the hippocampus and neocortex [81, 82].

Finally, in the domain of error correction and metacognition, the system's built-in feedback mechanism and the LLM's ability to provide regulated, fine-tuned responses align with the brain's natural error detection and reinforcement learning mechanisms. This involves the Anterior Cingulate Cortex (ACC), which plays a critical role in detecting cognitive conflict and signaling the need for correction [83, 84, 85]. Simultaneously, dopaminergic reward circuits [86, 87] reinforce correct understanding, as the system provides immediate and positive feedback. This immediate feedback loop fosters metacognitive awareness, enabling learners to refine their understanding and adapt their approach to problems. The system's transparency about response sources also aids in this metacognitive self-assessment.

3.5.4. Addressing Misuse and Fostering Ethical AI Use

A significant concern within the educational landscape is the potential for LLM misuse, including plagiarism and over-reliance. OWLLM's design principles directly confront this challenge. The system's ability to configure commands and fine-tune responses is specifically aimed at shaping the AI's behavior within an ethical educational context, thereby mitigating dishonest practices. The framework of human-AI collaboration, as proposed by Kasneci et al. (2023) [56], is integral to OWLLM's philosophy, emphasizing the importance of standardization, transparency, and reflective practice. This approach promotes responsible use by encouraging students to critically assess AI assistance and to recognize where human judgment, creativity, and contextual understanding remain indispensable. By seamlessly integrating the LLM in close proximity to course materials and focusing on supplementing rather than replacing core learning, OWLLM aims to redirect students towards productive engagement with the material, fostering a deeper understanding rather than merely enabling circumvention of academic responsibilities. This strategic implementation allows for the harnessing of AI's educational potential within a robust pedagogical ecosystem that upholds student agency and ethical responsibility.

In summation, OWLLM's design transcends a mere technical implementation; it is a meticulously crafted prototype that aims to align with and bolster core psychological and neurocognitive

processes inherent to learning. By intentionally addressing motivational needs, judiciously managing cognitive load, and supporting the brain's natural mechanisms for information processing, OWLLM endeavors to establish a more effective, engaging, and ethically responsible AI-assisted educational environment.

4. Design and Implementation

The design and implementation outlines the web application's foundational elements. This involves defining user needs and features with user stories, showcasing user interfaces and page layouts, as well as presenting an entity relationship diagram for overall database structure. These elements work together to effectively propose and illustrate the core application concept. The design process follows a user-centered approach, beginning with comprehensive user story development that captures the relative needs of students, educators, and administrators within the contextual use of the application. The implementation strategy emphasizes modular architecture, promoting scalability and maintainability while integrating advanced AI capabilities through a Retrieval Augmented Generation (RAG) system. The user interface design prioritizes simplicity in accessibility and intuitive navigation across the role-based access. The database architecture leverages PostgreSQL with pgvector extensions to support vector embedding, and semantic search capabilities, while the application architecture implements RESTful APIs to facilitate seamless communication between frontend and backend components such as the Open AI API's for generating the embeddings and providing LLM functionality, and or the general purpose Express API routes for dynamic database interaction and functionality. This section provides a detailed examination of how these design decisions translate into functional software components, demonstrating the practical application of theoretical concepts in developing an application aligned to serve contextual objectives in education. Furthermore, this section serves as comprehensive documentation of the system, providing detailed technical specifications, architectural decisions, and implementation details that would aim to guide any consideration for future development, maintenance, and potential deployment in legitimate educational environments.

4.1. Project Organization and Development Approach

The OWLLM project follows a well-structured organization that promotes maintainability, scalability, and clear separation of concerns. The project utilizes modern AI-assisted development tools, particularly Cursor IDE, to accelerate development of common web application functionality while focusing human effort on the unique AI integration challenges that require deep understanding of the RAG system and component interactions.

4.1.1. File Structure and Organization

The OWL-LM project follows a well-structured, role-based architecture that clearly separates concerns and facilitates maintainable development. The project is organized into a hierarchical structure with distinct frontend, backend, and configuration components.

4.1.1.a. Root Level Organization

- **Main Application Directory (`owl11m/`)** - Contains the complete application with clear separation between frontend and backend components.
- **`README.md`** - Project documentation and setup instructions.
- **Version Control** - Git repository with appropriate `.gitignore` and `.gitattributes` configurations.

4.1.1.b. Backend Architecture (`owl11m/backend/`)

The backend follows a modular Node.js/Express architecture with a clear separation of responsibilities.

4.1.1.c. Core Server Files

- **index.js** - Main Express server entry point and middleware configuration.
- **package.json & package-lock.json** - Node.js dependencies and project configuration.
- **db.js** - Database connection and configuration management.

4.1.1.d. API Layer (*api/*)

- **adminAPI.js** - Administrative endpoints for system management (80KB, 2,523 lines).
- **educatorAPI.js** - Educator-specific functionality and course management (128KB, 3,921 lines).
- **studentAPI.js** - Student-facing features and learning interactions (55KB, 1,776 lines).

4.1.1.e. Utility Modules (*utils/*)

- **llmUtils.js** - AI/LLM integration and processing logic (40KB, 970 lines).
- **supabase.js** - Database operations and Supabase client configuration (8.1KB, 255 lines).
- **validation.js** - Input validation and data sanitization (10KB, 390 lines).
- **upload.js** - File upload handling and processing (2.1KB, 92 lines).
- **server-check.js** - Health monitoring and system diagnostics (990B, 40 lines).

4.1.1.f. Database Management (*sql/*)

- **database.sql** - Complete database schema definitions (109KB, 3,433 lines).
- **database-sample-data.sql** - Initialization data and sample records (154KB, 2,988 lines).

4.1.1.g. Development and Testing (*debug/*)

- **YouTube Integration Testing** - Multiple scripts for video processing validation.
- **RAG System Diagnostics** - Retrieval-Augmented Generation testing and debugging.
- **API Endpoint Testing** - Comprehensive endpoint validation scripts.
- **File Upload Testing** - Various upload scenarios and format handling.
- **LLM Function Testing** - AI integration validation and parameter testing.
- **System Configuration Testing** - Bucket configuration and settings validation.

4.1.1.h. Frontend Architecture (*owllm/frontend/*)

The frontend implements a role-based user interface system with separate components for each user type.

4.1.1.i. Shared Components

- **pages/index.html** - Main landing page and navigation hub (27KB, 441 lines).
- **pages/login.html** - Authentication interface (3.3KB, 78 lines).
- **scripts/index.js** - Core frontend functionality (4.3KB, 169 lines).
- **scripts/login.js** - Authentication logic (11KB, 354 lines).
- **styles/index.css** - Global styling and shared components (10KB, 630 lines).
- **styles/login.css** - Authentication-specific styling (5.9KB, 307 lines).

4.1.1.j. Role-Specific Interfaces

Each role (admin, educator, student) has dedicated directories containing:

- **HTML Pages** - Role-specific user interfaces and functionality.
- **JavaScript Scripts** - Client-side logic and API interactions.
- **CSS Stylesheets** - Role-specific styling and theming.

4.1.1.k. Administrator Interface (*admin/*)

- Dashboard, analytics settings, course management, user management (schools, educators, students).
- Profile management and system configuration.

4.1.1.l. Educator Interface (*educator/*)

- Course content management, student engagement tracking, assignment creation.
- Notes and comments system, question-answer management, student monitoring.

4.1.1.m. Student Interface (*student/*)

- Course enrollment, content consumption, note-taking capabilities.
- Question submission, profile management, and the Curiotron learning interface.

4.1.1.n. Architectural Benefits

This organization provides several key advantages:

- **Separation of Concerns** - Clear boundaries between frontend presentation, backend logic, and data management.
- **Role-Based Access** - Isolated interfaces prevent unauthorized access and simplify permission management.
- **Modular Development** - Independent development and testing of components without affecting other parts.
- **Scalability** - Easy addition of new features or user roles without disrupting existing functionality.
- **Maintainability** - Logical grouping makes code location and modification straightforward.
- **Testing Infrastructure** - Comprehensive debug directory enables thorough system validation.

The structure demonstrates a mature understanding of software architecture principles, with clear integration points between frontend and backend components while maintaining the flexibility for independent development and deployment.

4.1.2. AI-Assisted Development Implementation

The implementation process leveraged AI code-building tools, particularly Cursor IDE, to accelerate development of standard web application functionality. This approach allowed the development team to focus on the unique challenges of LLM integration and RAG system implementation rather than spending time on common patterns like authentication, CRUD operations, and basic UI components.

AI tools were particularly valuable for:

- Generating boilerplate code for API endpoints and database operations
- Creating standard authentication and authorization patterns
- Developing common UI components and responsive layouts
- Implementing error handling and validation logic

However, the core AI integration components required significant human expertise and understanding of:

- RAG system architecture and vector embedding concepts
- OpenAI API integration and parameter optimization
- Document processing pipelines and chunking strategies
- Semantic search implementation and similarity algorithms

The development process involved iterative refinement of AI-generated code to ensure proper integration with the project's specific architecture and requirements. This hybrid approach of AI-assisted development for common functionality combined with human oversight for AI integration components proved highly effective for this educational technology project.

4.2. User Stories

User stories are concise, plain-language descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically

follow a simple template: "As a [type of user], I want [an action] so that [a benefit/value]." User stories and their associated use cases are instrumental in laying out the features and functionality of the web application. They provide a clear understanding of what users need to accomplish and why those actions are important to them.

The list below outlines the core interactions currently implemented for students, educators, and administrators. While all primary functionalities have been addressed in the current version of OWLLM, additional supplemental features—especially those supporting more complex integrations—remain possible for future development. These are not detailed here but are considered for later iterations based on institutional needs and system feedback.

4.2.1. User Stories by Category

The following user stories are grouped by common functionalities and roles. Each story includes the type of user, the action, the rationale behind the action, and a reference to the corresponding user interface figure(s) where implemented.

4.2.1.a. Account Management (Student + Educator + Admin)

- A1. As a student, I should be able to log in using my email and password, so that I can access the main functional components of the application for course interaction. (Figure 1)
- A2. As an educator, I should be able to log in using my email and password, so that I can access the main functional components of the application for course management and student oversight. (Figure 1)
- A3. As an admin, I should be able to log in using my username and password, so that I can access the administrative panel for system management. (Figure 2)
- A4. As a student, when logged in, I should be able to access my profile data, so that I can manage my login credentials and profile information. (Figure 11)
- A5. As an educator, when logged in, I should be able to access my profile data, so that I can manage my login credentials. (Figure 11)
- A6. As an admin, when logged in, I should be able to access my profile data, so that I can manage my login credentials. (Figure 11)
- A7. As a student, when accessing my profile data, I should be able to update my email and password, so that I can maintain the security of my account. (Figure 11)
- A8. As a student, when accessing my profile data, I should be able to include and/or update my profile information that consists of my major, minor, research interests, and career plans/goals so as to later provide LLM context for relevant responses. (Figure 11)
- A9. As an educator, when accessing my profile data, I should be able to update my email and password, so that I can maintain the security of my account. (Figure 11)
- A10. As an admin, when accessing my profile data, I should be able to update my email and password, so that I can maintain the security of my account. (Figure 11)

4.2.1.b. Course Access + Navigation (Student + Educator)

- B1. As a student, when logged in, I should be able to access a list of courses I am enrolled in (preconfigured in DB), so that I can easily find and navigate to a course. (Figure 4, Figure 5)
- B2. As a student, when accessing the list of courses, I should be able to select one, so that I can enter a specific course environment for interaction. (Figure 5, Figure 6)
- B3. As a student, after selecting a course, I should be directed to the main course page if I am enrolled in that course, so that I can begin interacting with the course. (Figure 6)
- B4. As a student, once on the main course page, I should be able to view and select a course material. (Figure 7)
- B5. As an educator, when logged in, I should be able to access a list of courses I have been assigned to teach (preconfigured in DB), so that I can easily find and navigate to a course. (Figure 5)
- B6. As an educator, when accessing the list of courses, I should be able to select one, so that I

- can enter a specific course environment for managing and interaction. (Figure 5)
- B7. As an educator, after selecting a course, I should be directed to the main course page if I am assigned to teach that course, so that I can begin managing and interacting with the course. (Figure 6)
- B8. As an educator, once on the main course page, I should be able to view and select a course material. (Figure 7)

4.2.1.c. Student: Content Interaction + LLM Tool

- C1. As a student, when interacting with the course material section, I should be able to select a course material from a list of supported formats and have it displayed or played in a viewer/player component, as well as have the ability to download a copy of the material. (Figure 6, Figure 7)
- C2. As a student, when interacting with a selected material, I should be able to save notes and questions I have on the material by typing into an assistive LLM tool using the command triggers \note or \question followed by the content. (Figure 7)
- C3. As a student, I should be able to type any question into the assistive LLM tool, and have it respond with a regulated, fine-tuned response. (Figure 7)
- C4. As a student, I should be able to view, update, and delete my saved notes and questions for a course. (Figure 8, Figure 9)
- C5. As a student, I should be able to filter and search my notes and questions using keywords or by selecting a specific material from a dropdown. (Figure 8, Figure 9)
- C6. As a student, I should be able to use the \relevance command to ask how course content relates to my academic and career interests. (Figure 7, Figure 10)
- C7. As a student, I should be able to use the \feedback command to provide feedback about the AI system. (Figure 18, Figure 19)
- C8. As a student, I should be able to view my saved relevance responses in a dedicated section. (Figure 10)

4.2.1.d. Educator: Course Management + Student Oversight

- D1. As an educator, I should be able to upload course materials in supported formats to an organized selectable list on the course material section page. (Figure 12)
- D2. As an educator, I should be able to view or play a selected course material in a viewer/player component and download it. (Figure 6)
- D3. As an educator, I should be able to view a list of students enrolled in my course. (N/A – Figure not included in current set)
- D4. As an educator, I should be able to view individual student profiles and see their notes and questions for the course. (Figure 8, Figure 9)
- D5. As an educator, I should be able to view and respond to questions students asked me. (Figure 9)
- D6. As an educator, I should be able to use the LLM tool to ask questions about course content. (Figure 7)
- D7. As an educator, I should be able to use the \feedback command to provide feedback about the AI system. (Figure 18, Figure 19)

4.2.1.e. Admin: System Management + Analytics

- E1. As an admin, I should be able to manage schools by adding, editing, and deleting school information. (Figure 14)
- E2. As an admin, I should be able to manage students by adding, editing, and deleting student accounts. (Figure 12)
- E3. As an admin, I should be able to manage educators by adding, editing, and deleting educator accounts. (N/A – Figure not included in current set)
- E4. As an admin, I should be able to manage courses by adding, editing, and deleting course

- information. (N/A – Figure not included in current set)
- E5. As an admin, I should be able to manage course enrollments by adding and removing students from courses. (Figure 15)
 - E6. As an admin, I should be able to configure LLM settings including model parameters, embedding models, and system prompts. (Figure 16)
 - E7. As an admin, I should be able to view and analyze feedback from students and educators. (Figure 18)
 - E8. As an admin, I should be able to enable or disable specific features like relevance analysis. (Figure 16)

4.3. Core System Architecture

The OWLLM system implements a modular architecture that integrates AI capabilities with traditional web application components. The architecture focuses on the key LLM integration and interaction functionality while maintaining clean separation between frontend, backend, and data layers.

4.3.1. Core Architectural Diagram

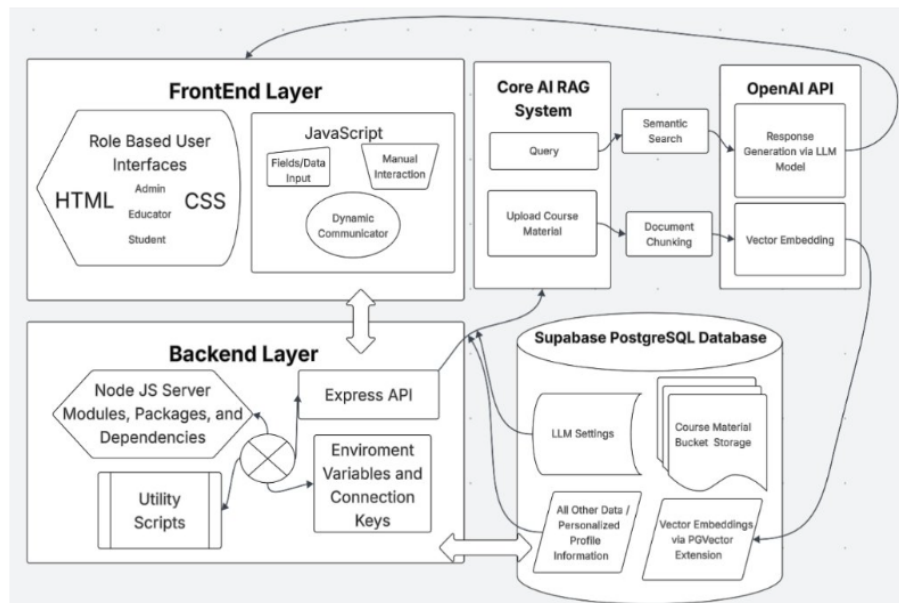


Fig. 1. Core Architectural Component Diagram

A diagram of the core architectural components, with a focus on the AI RAG system and OpenAI integration.

4.3.2. Key Architectural Components

The system consists of four main layers:

- **Frontend Layer** - Role-based interfaces for students, educators, and administrators built with HTML, CSS, and JavaScript
- **Backend Layer** - Node.js/Express server handling API requests, authentication, and AI integration
- **Database Layer** - PostgreSQL database with pgVector extension for vector embeddings and semantic search, and the built in storage bucket for document storage
- **RAG System/LLM Layer** - OpenAI API for the base LLM model, and vector embedding model

4.3.3. Pre-LLM Integration MIME Type Compatibility

Before any document processing in the RAG system, there is a pre-document processing phase of file identification and MIME type extension compatibility gauging. The extension of the file, and or the MIME type, is a indication of how that file's data is formatted. All such files listed below are supported, requiring specific libraries, dependencies, and algorithms to properly process them for their text content:

- **.txt** - Native Node.js Buffer processing (buffer.toString("utf-8"))
- **.pdf** - pdf-parse library
- **.doc** - LibreOffice conversion + pdf-parse library
- **.docx** - mammoth library
- **.pptx** - LibreOffice conversion + pdf-parse library
- **.xlsx** - xlsx library
- **.csv** - Native Node.js Buffer processing (buffer.toString("utf-8"))
- **.mp3** - OpenAI Whisper API (openai library)
- **.mp4** - FFmpeg (extractAudioFromVideo) + OpenAI Whisper API
- **.py** - Native Node.js Buffer processing (buffer.toString("utf-8"))
- **.java** - Native Node.js Buffer processing (buffer.toString("utf-8"))
- **.html** - Native Node.js Buffer processing (buffer.toString("utf-8"))
- **.css** - Native Node.js Buffer processing (buffer.toString("utf-8"))
- **.js** - Native Node.js Buffer processing (buffer.toString("utf-8"))
- **.sql** - Native Node.js Buffer processing (buffer.toString("utf-8"))

4.3.4. LLM Integration Architecture

The application and its core AI functionality centers around the RAG (Retrieval-Augmented Generation) system and its different parts:

- **Document Processing** - Uploaded materials and their text contents are chunked and converted to vector embeddings using OpenAI's text-embedding-3-small model
- **Semantic Search** - User queries are embedded and matched against stored document chunks using cosine similarity
- **Response Generation** - Relevant chunks are combined with user queries and sent to GPT-4 for context-aware responses
- **Command Processing** - Special commands (\note, \question, \relevance, \feedback) trigger specific system behaviors

4.4. User Manual and System Interaction Guide

This section provides a comprehensive guide for users to interact with the OWLLM system.

4.4.1. Getting Started

To access the OWLLM system, users must first log in through the authentication portal. The system supports three distinct user roles, each with different access levels and capabilities:

- **Students** - Access enrolled courses, interact with materials, and use AI assistance
- **Educators** - Manage course content, monitor student engagement, and configure course settings
- **Administrators** - System-wide management, user administration, and AI system configuration

4.4.2. Student Interface and Interactions

Students begin their experience by selecting their enrolled courses from the course dashboard. Once in a course, students can:

- **View Course Materials** - Access uploaded documents, videos, and other educational content

- **AI Chat Interface** - Interact with the LLM tool for questions, notes, and content exploration
- **Command System** - Use specific commands for different types of interactions:
- **Personal Workspace** - Manage saved notes, questions, and relevance analyses
 - \note [content] - Save personal notes about course materials
 - \question [content] - Ask questions about specific content
 - \relevance - Analyze how course content relates to career goals
 - \feedback - Provide system feedback and ratings

4.4.3. Educator Interface and Interactions

Educators have access to course management tools and student oversight capabilities:

- **Course Content Management** - Upload and organize educational materials in various formats
- **Student Engagement Monitoring** - View student notes, questions, and interaction patterns
- **AI Assistance** - Use the LLM tool for content preparation and student support
- **Q&A Management** - Respond to student questions and provide clarifications

4.4.4. Administrator Interface and Interactions

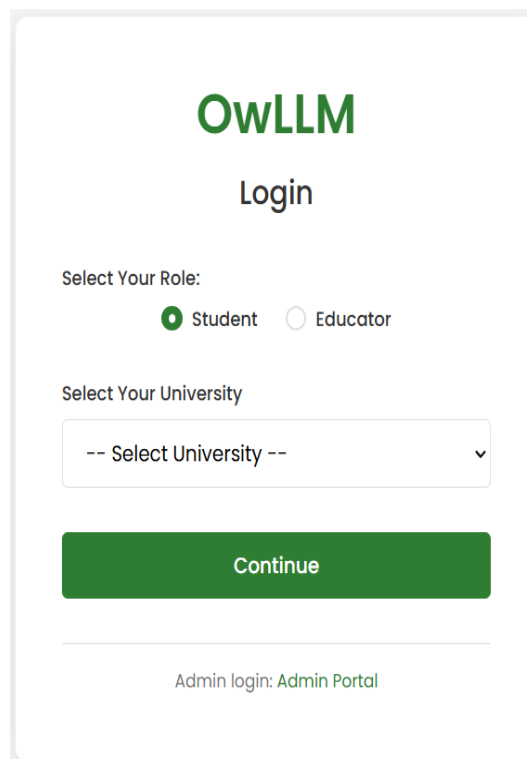
Administrators oversee the entire system and configure AI behavior:

- **User Management** - Create and manage student and educator accounts
- **Institutional Management** - Configure schools and course structures
- **AI System Configuration** - Adjust LLM parameters, embedding models, and system prompts
- **Analytics and Feedback** - Monitor system usage and user satisfaction

4.5. User Interface and Interactions

The OWLLM system features a clean, intuitive interface designed to facilitate seamless interaction between users and AI capabilities. The interface prioritizes accessibility and role-based functionality while maintaining visual consistency across all user types.

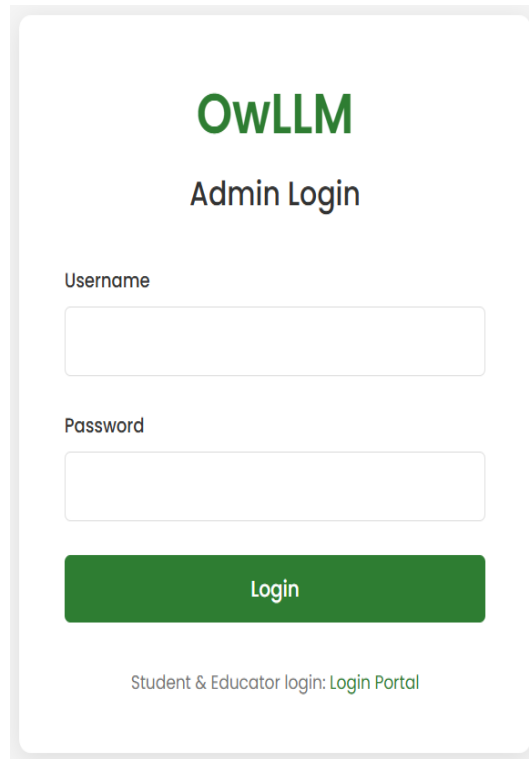
4.5.1. Authentication and Navigation



The image shows a login interface for OwLLM. At the top, the text "OwLLM" is displayed in a large, bold, green font, followed by "Login" in a smaller, black font. Below this, there is a section titled "Select Your Role:" with two radio button options: "Student" (which is selected, indicated by a green dot) and "Educator" (which is unselected, indicated by a grey dot). Underneath, there is a section titled "Select Your University" with a dropdown menu that currently shows "-- Select University --" and a downward arrow. Below the dropdown is a large, green, rounded rectangular button with the text "Continue" in white. At the bottom of the interface, there is a link that says "Admin login: Admin Portal" in a small, green font.

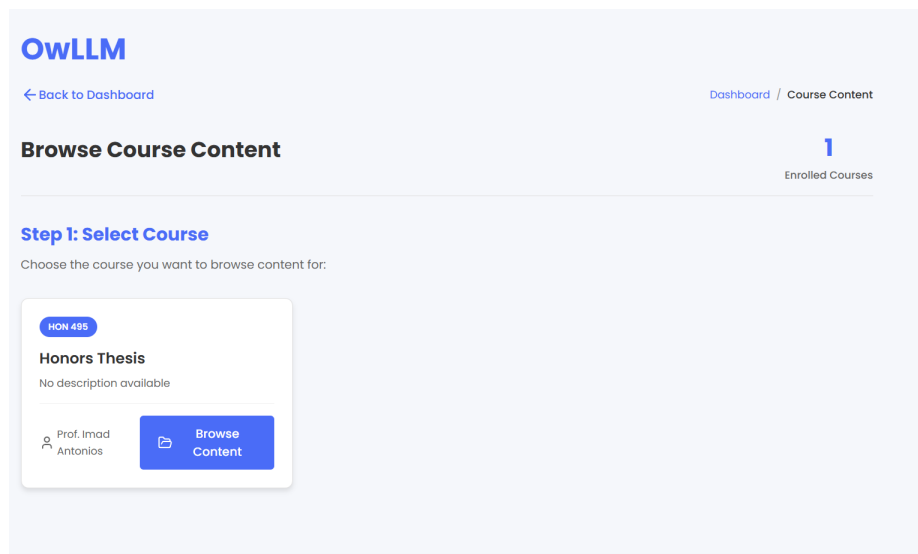
Fig. 2. Student and Educator Login Interface-Supports user stories: A1, A2

The unified login page provides role-based authentication for both students and educators, with institution selection and secure credential entry. The interface adapts based on the selected user role.



The image shows a login interface for OwLLM. At the top, the text "OwLLM" is displayed in a large, bold, green font. Below it, "Admin Login" is written in a smaller, black font. There are two input fields: "Username" and "Password", both with light gray borders. Below the password field is a green "Login" button with white text. At the bottom, there is a link that says "Student & Educator login: Login Portal" in a small, gray font.

Fig. 3. Administrator Login Interface-Supports user story: A3
Dedicated administrator login interface with enhanced security measures and system-level access controls specific to administrative functions.



The image shows a course selection dashboard for OwLLM. At the top left, the "OwLLM" logo is in blue. Below it is a link "← Back to Dashboard". On the top right, there is a breadcrumb "Dashboard / Course Content". The main heading is "Browse Course Content". To the right of this heading is a blue box with the number "1" and the text "Enrolled Courses" below it. Below the heading is a section titled "Step 1: Select Course" with the instruction "Choose the course you want to browse content for:". There is a card for a course titled "Honors Thesis" with the code "HON 495" in a blue box. Below the title is the text "No description available". At the bottom left of the card is a profile icon and the name "Prof. Imad Antonios". At the bottom right is a blue button with a folder icon and the text "Browse Content".

Fig. 4. Course Selection Interface-Supports user stories: B5, B6
The course selection dashboard displays available courses with search and filtering capabilities. This interface is similar for both students and educators, though educators see manage course content instead of just browse.

4.5.2. Student Interface and AI Interactions

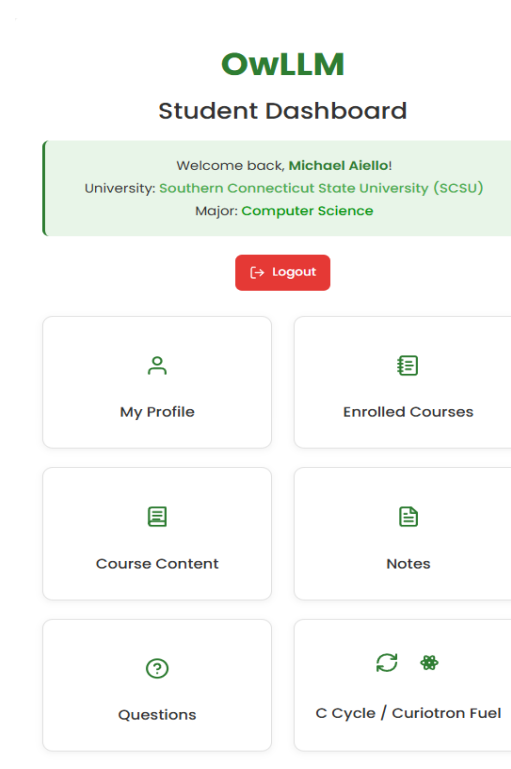


Fig. 5. Student Dashboard-Supports user story: B1

The student dashboard provides quick access to the main components of the application.

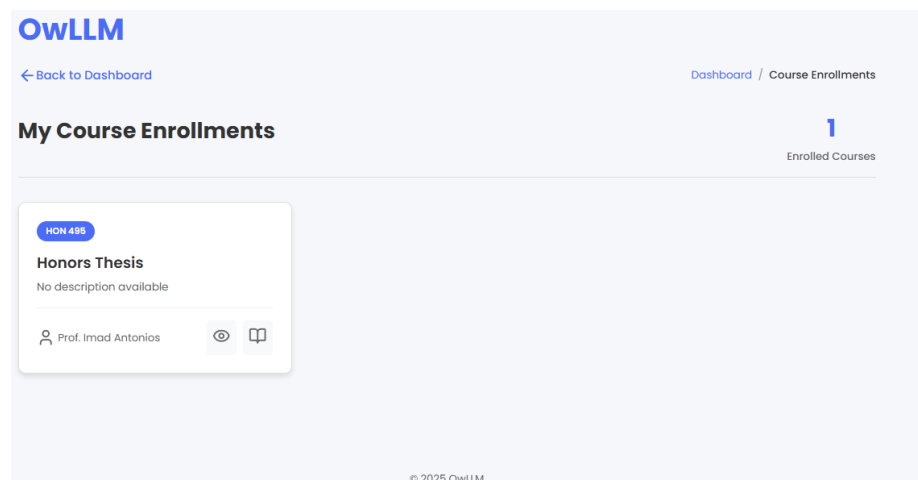


Fig. 6. Student Course Enrollments-Supports user stories: B1, B2

Students can view their course enrollments, with quick access to the course contents or course information.

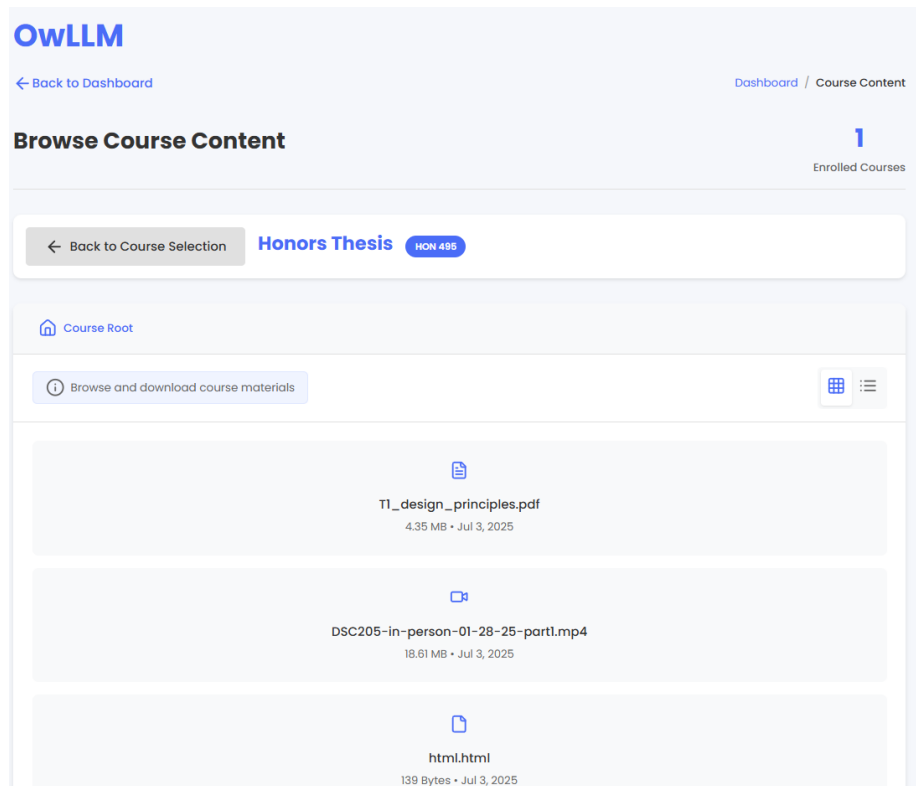


Fig. 7. Student Course Content Browser-Supports user stories: B4, B8
Students can browse and search through course materials. The browse interface is the same for educators and students; however, educators have an upload course content feature.

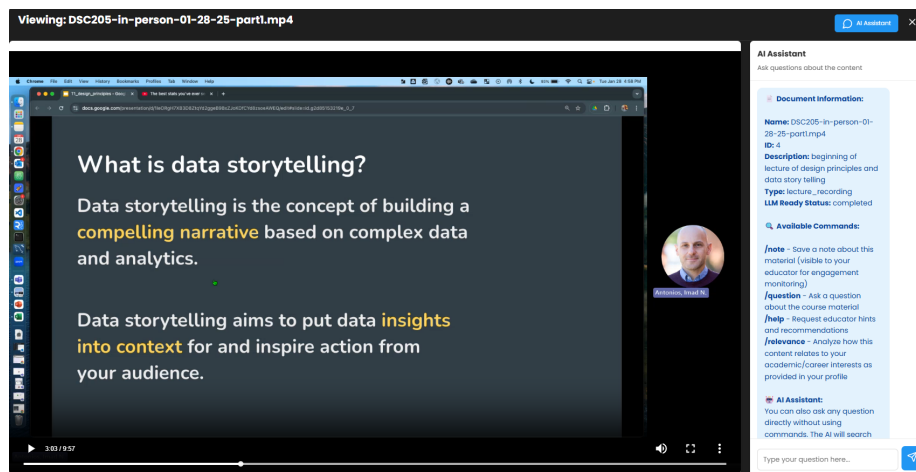


Fig. 8. Student Course Material Viewer / AI Assistant-Supports user stories: B3, C1, C2, C3, D2, D6
Interactive course material viewer with AI chat integration, allowing students to ask questions and receive contextual responses based on the specific content being viewed.

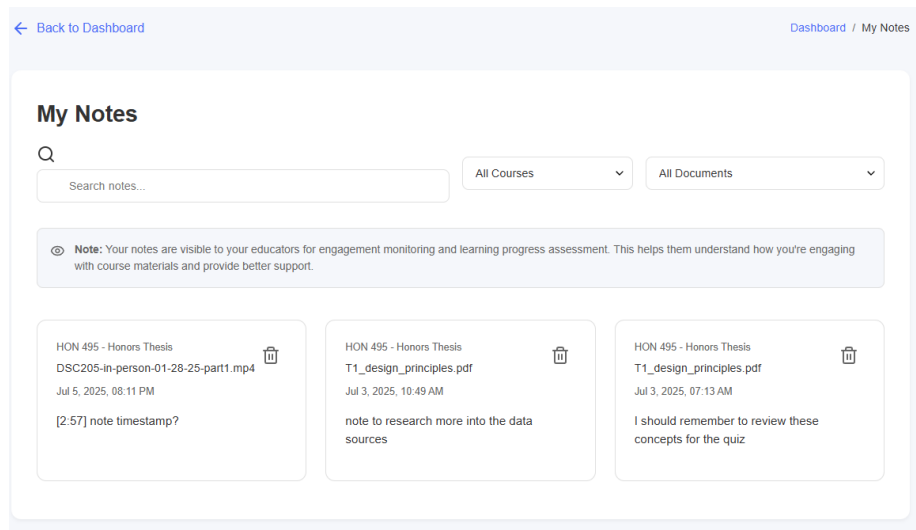


Fig. 9. Student Notes Management-Supports user stories: C4, C5, D4
Students can view, edit, and organize their saved notes. This feature is specific to students; however, educators have the capacity to view a student's notes.

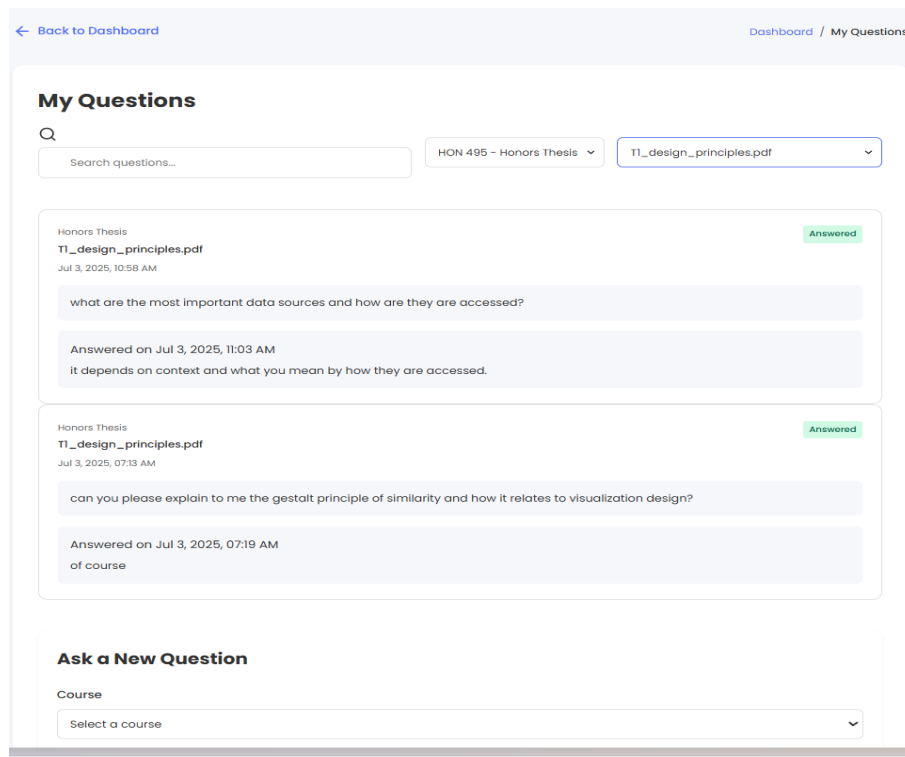


Fig. 10. Student Questions Management-Supports user stories: C4, C5, D5
Students can manage their saved questions and see the answers or responses from educators.

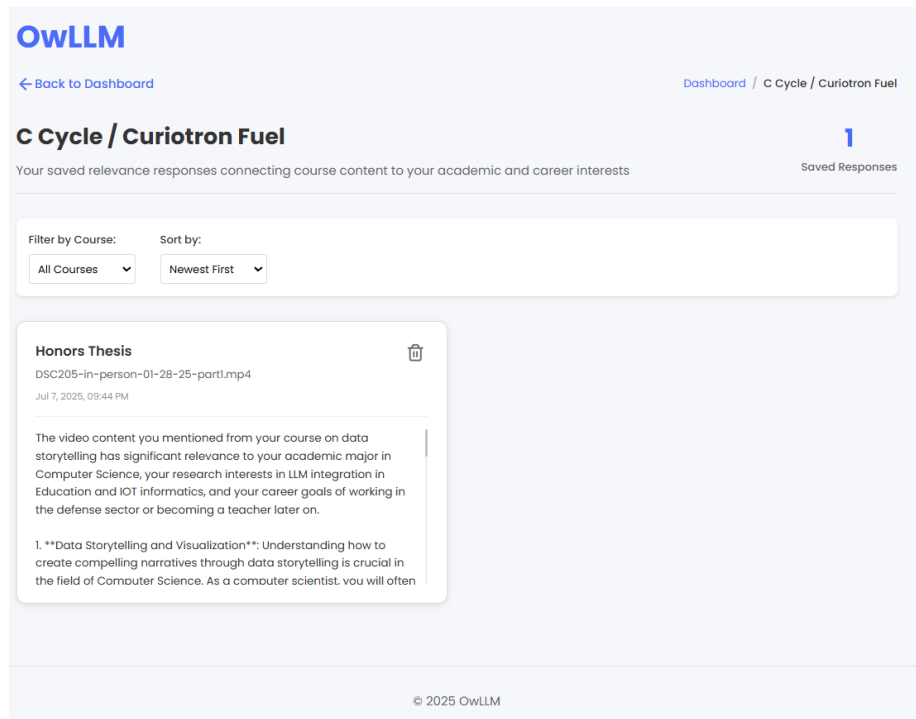


Fig. 11. Student Relevance Analysis-Supports user stories: C6, C8
AI-powered relevance analysis interface that connects course content to students' academic and career interests, providing personalized learning insights and recommendations.

OwLLM

[← Back to Dashboard](#)[Dashboard / Student Profile](#)

Student Profile



Michael Aiello
aiellom2@southernct.edu

First Name
Michael

Last Name
Aiello

Email
aiellom2@southernct.edu

Phone
2038954666

University
Southern Connecticut State University

Major
Computer Science

Minor
Honors College

Research Interests
LLM Integration in Education and IOT Informatics

Career Goals
Work and gain experience at companies or in a defense sector to maybe become a teacher later

Account Created
June 27, 2025

Update Profile Information

First Name

Last Name

Email

Phone

Major

Minor

Research Interests

Career Goals

Current Password

New Password (leave blank to keep current)

Confirm New Password

© 2025 OwLLM

Fig. 12. Student Profile Information-Supports user stories: A4, A5, A6, A7, A8, A9, A10
Student profile management interface where users can update personal information, academic interests, and learning preferences. Similar profile interfaces exist for educators and administrators with role-specific fields.

4.5.3. Educator Interface and AI Tools

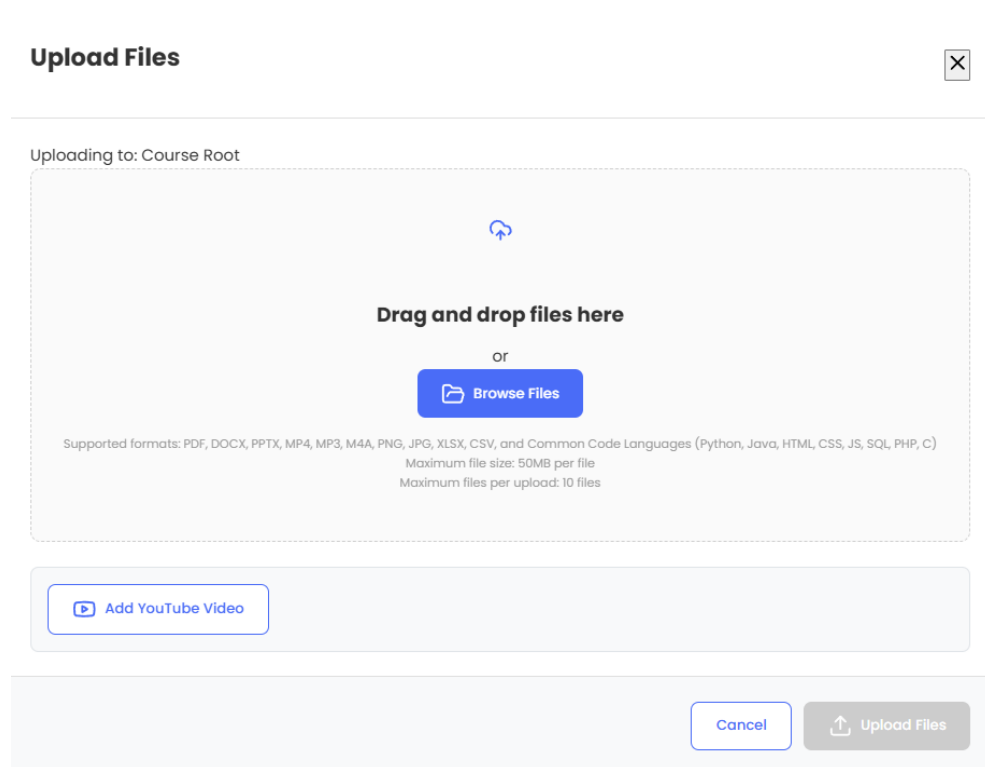


Fig. 13. Educator Course Content Upload-Supports user story: D1
Multi-format content upload interface with automatic AI processing and vector embedding generation. This functionality is specific to educators and includes advanced content management tools not available to students.

4.5.4. Administrative Interface and AI Configuration

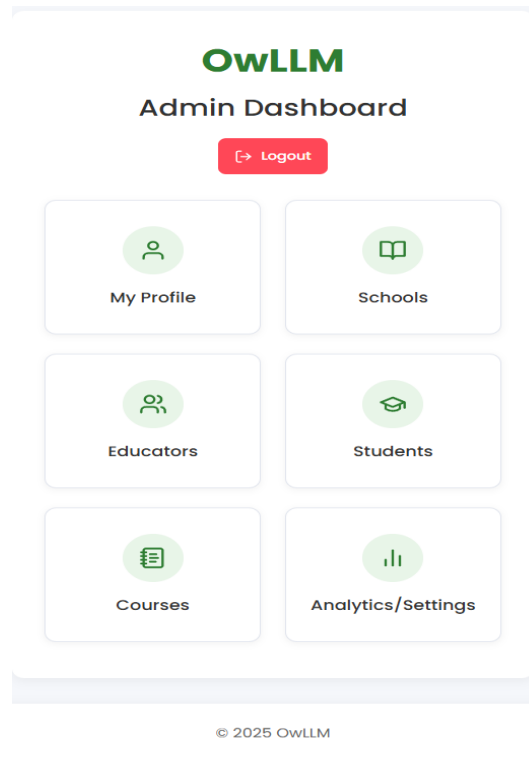


Fig. 14. Administrative Dashboard(No specific user stories currently mapped)
This dashboard is specific to administrators and provides quick access to system-level management features and settings controls.

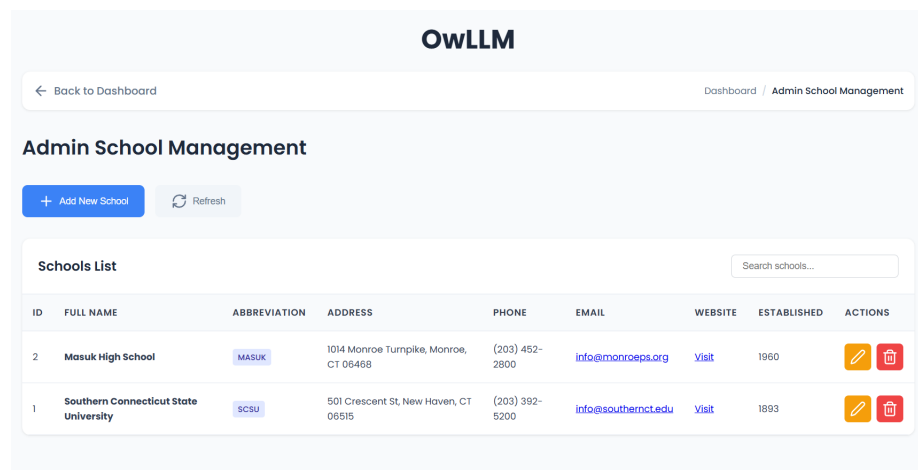


Fig. 15. Administrator School Management-Supports user story: E1
School and institution management interface for creating and managing educational institutions within the system. This administrative function is specific to system administrators.

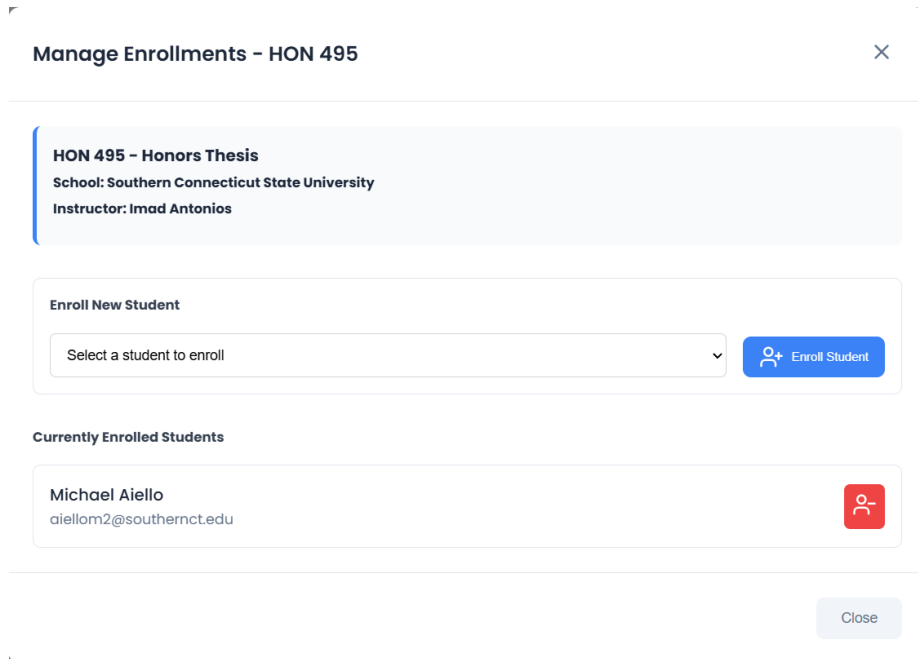


Fig. 16. Administrator Enrollment Management-Supports user story: E5
System-wide enrollment management tools for overseeing student course enrollments across all institutions. This administrative oversight capability is specific to system administrators.

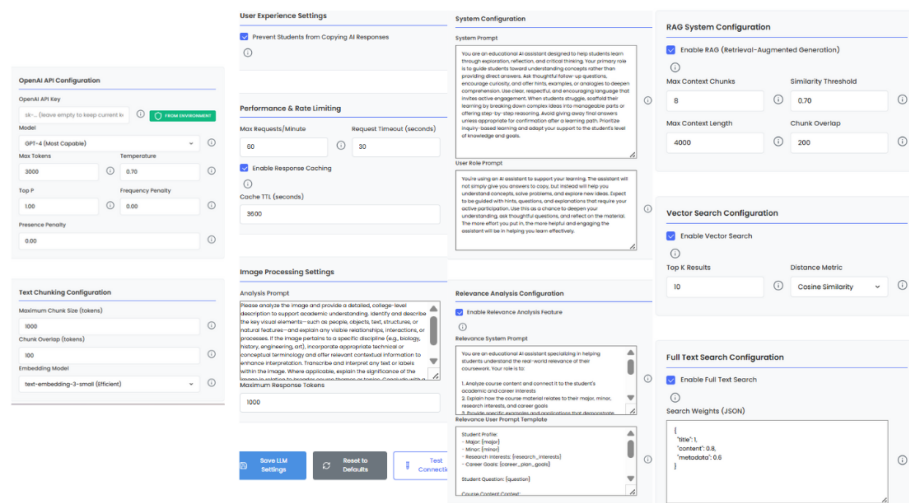


Fig. 17. AI System Configuration-Supports user stories: E6, E9
Advanced settings panel for customizing AI behavior, embedding models, and system parameters. This configuration interface is specific to administrators and provides system-level AI customization.

Document Type Management

Document Types

Manage the types of documents that can be uploaded to courses.

+ Add New Type

Label	Value	Description	Status	Actions
 Administrative Documents	administrative_documents	Records, reports, and administrative content	Active	 
 Accommodation Plan	accommodation_plan	Student accommodation or accessibility plan	Active	 
 Attendance Record	attendance_record	Class attendance documentation	Active	 
 Certification	certification	Course completion or achievement certificate	Active	 
 Disciplinary Notice	disciplinary_notice	Student conduct or disciplinary record	Active	 
 Grade Report	grade_report	Student grade or performance report	Active	 

Fig. 18. Document Type Configuration-Supports user story: E9
Administrative interface for configuring supported document types and their processing parameters. This system configuration is specific to administrators and affects how content is processed across the platform.

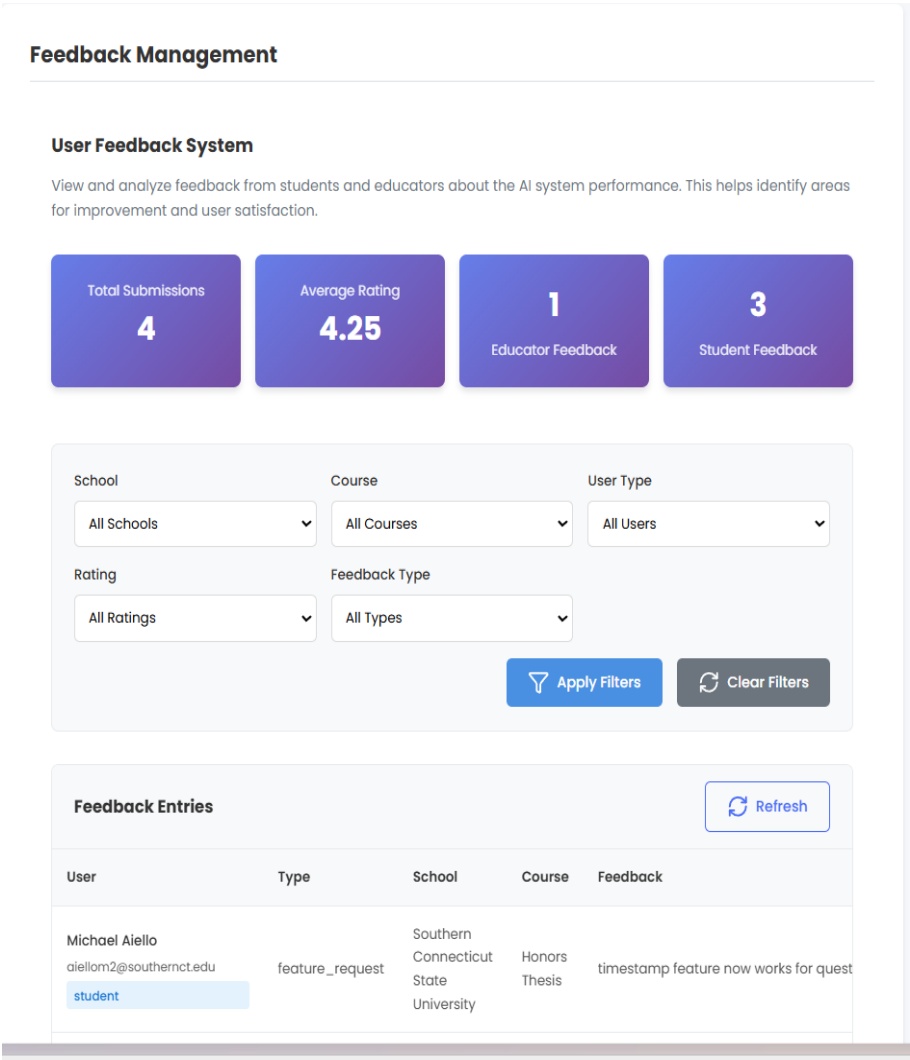


Fig. 19. Administrative Feedback Analytics-Supports user story: E7

System-wide analytics and feedback monitoring for AI performance and user satisfaction evaluation. This comprehensive analytics dashboard is specific to administrators and provides insights across all user types.

4.5.5. AI Command System and User Feedback

Share Your Feedback

×

Feedback Type:

Select feedback type... ▼

How would you rate your experience? (Optional):

★ ★ ★ ★ ★

Your Feedback:

Please share your thoughts, suggestions, or report any issues you encountered... 

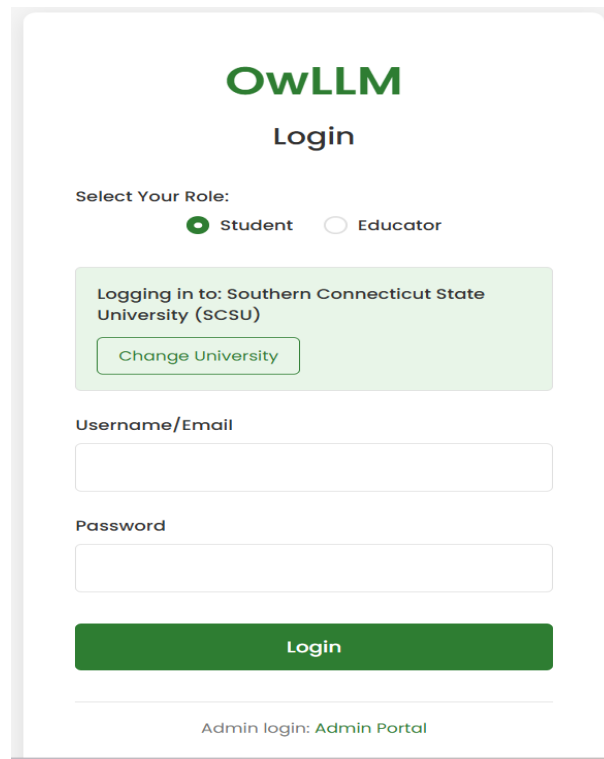
Skip Feedback

 Submit Feedback

Fig. 20. User Feedback System-Supports user stories: C7, D7, E7

Universal feedback collection and rating system available to all user types for continuous AI performance improvement. The interface adapts based on user role while maintaining consistent feedback collection mechanisms.

4.5.6. Institution-Specific Features



The image shows a login interface for OwLLM, specifically customized for Southern Connecticut State University (SCSU). The interface is clean and modern, with a white background and green accents. At the top, the OwLLM logo is displayed in green. Below it, the word "Login" is centered. A section titled "Select Your Role:" contains two radio buttons: "Student" (selected) and "Educator". Below this, a green box contains the text "Logging in to: Southern Connecticut State University (SCSU)" and a "Change University" button. The main login area has two input fields: "Username/Email" and "Password". A large green "Login" button is positioned below the input fields. At the bottom, there is a link for "Admin login: Admin Portal".

Fig. 21. SCSU-Specific Login Interface(No specific user stories currently mapped)
Institution-specific login interface demonstrating the system's ability to customize authentication and branding for different educational institutions while maintaining core functionality.

4.6. Database Architecture

The database design focuses on supporting the core AI functionality while maintaining data integrity and security. The schema emphasizes the relationships between users, courses, documents, and AI interactions, with comprehensive stored procedures that encapsulate business logic and ensure secure data access.

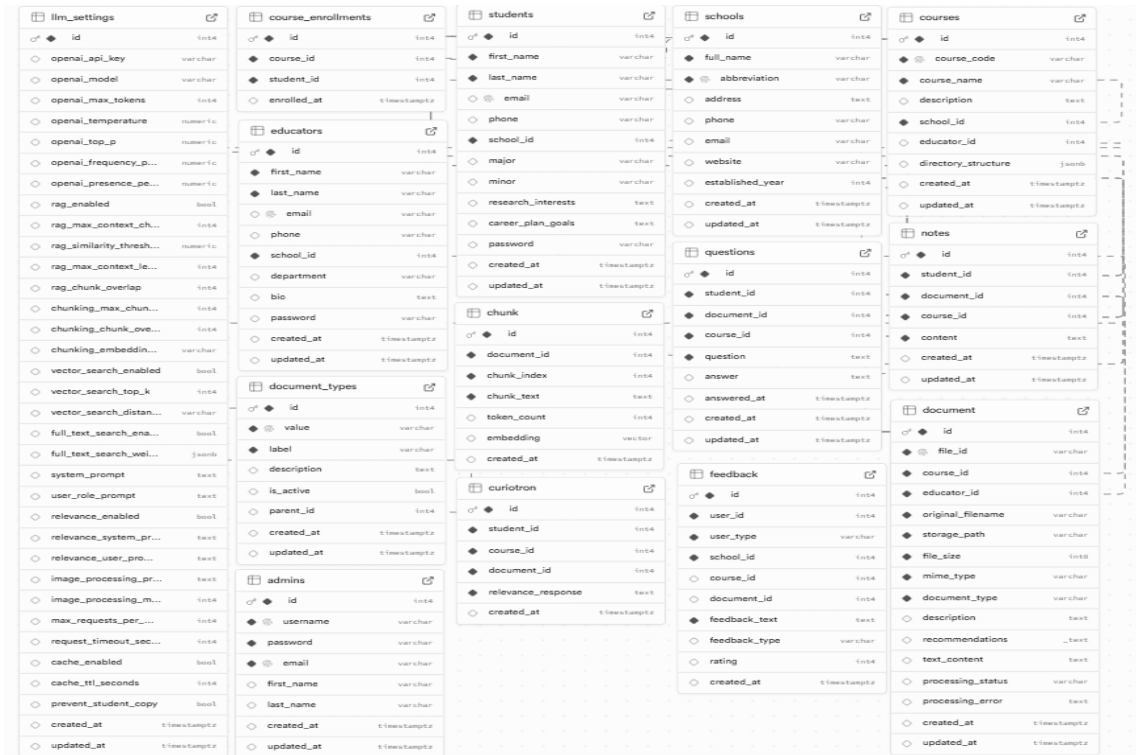


Fig. 22. Full Entity Relationship Diagram
Full ERD showing all entities and relationships.

4.6.1. Drop Cascade and Create or Replace Development Approach

During development, a "drop cascade and create or replace" approach was taken, which means that all tables, stored procedures, functions, and index constraints were deleted before recreating them. This approach allowed the full database schema to reside in one file and easily be updated. In a production environment or in a more comprehensive test database for development, a targeted, granular approach would be taken to protect the data, and build around or on-top of the current schema.

4.6.2. Core Database Tables and Relationships

The database schema is built around several interconnected tables that support the AI-powered educational platform. The complete database schema can be found in `owl1m/backend/sql/database.sql`, with sample data in `owl1m/backend/sql/database-sample-data.sql`.

4.6.2.a. User Management Tables

The `admins`, `educators`, and `students` tables implement role-based user management with institutional boundaries through the `schools` table. Each user type has specific authentication credentials and profile information, with students additionally storing academic interests and career goals to enable personalized AI interactions.

The `schools` table serves as the organizational foundation, containing institutional information and establishing the top-level boundary for data isolation. All users are associated with specific schools through foreign key relationships, ensuring proper institutional separation and access control.

4.6.2.b. Course and Enrollment Management

The `courses` table stores academic course information including unique course codes, descriptions, and instructor assignments. Each course is linked to a school and primary educator, with a `JSONB directory_structure` field that maintains hierarchical file organization for course materials.

The `course_enrollments` table manages the many-to-many relationship between students and courses, tracking enrollment dates and ensuring unique student-course combinations through database constraints. This table is crucial for determining access permissions to course materials and AI interactions.

4.6.2.c. Document Processing and AI Integration

The `document` table serves as the central repository for all educational materials, storing comprehensive metadata including file information, processing status, and extracted text content. Each document is linked to a specific course and educator, with support for various file formats and processing states.

The `chunk` table implements the document chunking strategy essential for RAG systems. Each document is split into manageable text segments (typically 1000 characters with 200-character overlap) with associated 1536-dimensional vector embeddings using the `pgvector` extension. This enables semantic search and context-aware AI responses.

The `document_types` table provides hierarchical categorization for educational materials, supporting parent-child relationships for detailed content classification and enabling sophisticated filtering and organization capabilities.

4.6.2.d. User Interaction and AI Feedback

The `notes` table allows students to create personal annotations on course materials, with each note linked to a specific student, document, and course. This supports personalized learning experiences while maintaining proper access controls.

The `questions` table facilitates student-instructor communication, tracking question status, answers, and response timestamps. This supports asynchronous learning interactions and provides educators with insights into student understanding.

The `feedback` table collects user feedback on AI system performance, including ratings, feedback types, and contextual information. This data supports system improvement and quality assurance processes.

The `curiotron` table stores AI-generated relevance analyses that connect course content to students' academic and career interests, enhancing student engagement by demonstrating the practical value of course materials.

4.6.2.e. System Configuration

The `llm_settings` table centralizes all AI system configuration parameters, including OpenAI API settings, RAG system parameters, vector search configurations, and user experience settings. The table enforces a single configuration record and supports comprehensive system customization.

4.6.3. Stored Procedures and Business Logic

The database includes an extensive collection of PostgreSQL stored functions that encapsulate business logic and provide secure, efficient data access patterns. These functions are designed with `SECURITY DEFINER` privileges and implement comprehensive access control through Row Level Security (RLS) policies. All stored procedures are defined in `owllm/backend/sql/database.sql`.

4.6.3.a. User Management Functions

The system provides comprehensive user management operations through stored procedures:

- `create_educator()` and `create_student()` - User registration with validation and institutional verification
- `get_educator_by_email()` and `get_student_by_email()` - Secure authentication with credential verification
- `update_educator_password()` and `update_student_password()` - Secure credential management with encryption
- `verify_educator_university()` and `verify_student_university()` - Institutional boundary validation

4.6.3.b. Course and Enrollment Functions

Course-related functions handle enrollment and administrative operations:

- `create_course()` - Course creation with validation for unique course codes and instructor assignment
- `enroll_student_in_course()` and `unenroll_student_from_course()` - Enrollment management with access control
- `get_course_students()` and `get_student_courses()` - Relationship queries with proper authorization
- `get_courses_by_educator()` and `get_students_by_educator()` - Instructor-specific data access

4.6.3.c. Document Processing and RAG Functions

Advanced document management functions support the AI processing pipeline:

- `create_document_record()` - Initial document registration with metadata extraction
- `update_document_processing_status()` - Processing state tracking for asynchronous operations
- `create_text_chunks()` - RAG system chunk generation with overlap management
- `get_documents_by_course()` - Content retrieval with access control
- `cleanup_orphaned_documents()` - Maintenance operations for data integrity

4.6.3.d. AI Interaction Functions

Functions supporting user interactions and AI system feedback:

- `create_note()` and `get_document_notes()` - Student annotation management
- `create_question()` and `answer_question()` - Q&A functionality with status tracking
- `create_feedback()` and `get_all_feedback()` - System evaluation and improvement
- `save_curiotron_response()` - Relevance analysis storage and retrieval

4.6.3.e. Configuration Management Functions

System configuration and maintenance functions:

- `get_llm_settings()` and `update_llm_settings()` - AI system configuration management
- `reset_llm_settings()` - System restoration to default parameters
- `test_llm_connection()` - System diagnostics and health checks
- `get_document_types()` - Content categorization and organization

4.6.4. RAG Search Implementation

The RAG (Retrieval-Augmented Generation) system is implemented through sophisticated vector search capabilities using PostgreSQL's pgvector extension. The system provides both semantic search and context-aware response generation. The core RAG functionality is implemented in `owlm/backend/utils/llmUtils.js`.

4.6.4.a. Vector Embedding Storage

The system utilizes 1536-dimensional vector embeddings generated by OpenAI's text-embedding-3-small model:

- **Chunk Generation** - Documents are processed into overlapping text segments of 1000 characters with 200-character overlap
- **Embedding Creation** - Each chunk is converted to a 1536-dimensional vector using OpenAI's embedding API
- **Storage Optimization** - Embeddings are stored using pgvector's efficient vector data type with IVFFlat indexing

4.6.4.b. Semantic Search Process

The RAG search process involves several key steps, implemented in the `searchDocuments()` function in `owllm/backend/utils/llmUtils.js`:

- 1) **Query Embedding** - User queries are converted to vector embeddings using the same model as document chunks
- 2) **Similarity Calculation** - Cosine similarity is computed between the query embedding and all stored document chunk embeddings
- 3) **Result Filtering** - Chunks are filtered based on configurable similarity thresholds (default 0.15)
- 4) **Top-K Retrieval** - The most relevant chunks (default 5) are selected for context generation
- 5) **Access Control** - Results are filtered based on user permissions and course enrollments

4.6.4.c. Search Configuration and Optimization

The RAG system includes configurable parameters for optimal performance, managed through the `llm_settings` table and accessed via `owllm/backend/utils/llmUtils.js`:

- **Similarity Thresholds** - Adjustable minimum similarity scores for result quality control
- **Top-K Settings** - Configurable number of chunks retrieved for context generation
- **Chunk Overlap Management** - Optimized overlap settings for context preservation
- **Model Selection** - Support for multiple embedding models (text-embedding-3-small, text-embedding-3-large, text-embedding-ada-002)

4.6.4.d. Context-Aware Response Generation

The retrieved chunks are integrated into the LLM response generation process through the `generateResponse()` function in `owllm/backend/utils/llmUtils.js`:

- **Context Assembly** - Relevant chunks are combined with the original user query
- **Prompt Engineering** - System prompts are customized based on user role and query type
- **Response Generation** - GPT-4 generates context-aware responses using the assembled information
- **Command Processing** - Special commands (`\note`, `\question`, `\relevance`, `\feedback`) trigger specific system behaviors

4.6.5. Security and Access Control

The database implements comprehensive security through Row Level Security (RLS) policies and stored procedures:

- **School-based Isolation** - Users can only access data from their institution
- **Role-based Access Control** - Different policies for students, educators, and administrators
- **Course-based Permissions** - Students can only access materials from enrolled courses
- **Data Ownership Policies** - Users can only access their own content and interactions

4.7. API Implementation and Code Structure

The backend API is implemented using Node.js and Express, with the main server file located at `owllm/backend/index.js`. The API routes are organized by user role and functionality:

4.7.1. Student API Routes

Student-specific functionality is implemented in `owllm/backend/api/studentAPI.js`:

- GET `/api/student/courses` - Retrieve enrolled courses
- GET `/api/student/course/:courseId/materials` - Access course materials
- POST `/api/student/chat` - AI chat interactions with command processing
- GET `/api/student/notes` - Retrieve personal notes
- POST `/api/student/notes` - Create new notes
- GET `/api/student/curiotron` - Access relevance analysis responses

4.7.2. Educator API Routes

Educator functionality is implemented in `owllm/backend/api/educatorAPI.js`:

- GET `/api/educator/courses` - Retrieve assigned courses
- POST `/api/educator/upload` - Upload course materials
- GET `/api/educator/students/:courseId` - View enrolled students
- GET `/api/educator/student/:studentId/notes` - Access student notes and questions
- POST `/api/educator/answer-question` - Respond to student questions

4.7.3. Admin API Routes

Administrative functionality is implemented in `owllm/backend/api/adminAPI.js`:

- GET `/api/admin/schools` - Manage institutional data
- POST `/api/admin/users` - Create and manage user accounts
- GET `/api/admin/courses` - Manage course structures
- GET `/api/admin/llm-settings` - Configure AI system parameters
- POST `/api/admin/llm-settings` - Update AI configuration
- GET `/api/admin/feedback` - Access system feedback and analytics

4.7.4. Utility Functions and Database Integration

Database connectivity and utility functions are implemented in several key files:

- `owllm/backend/utils/supabase.js` - Database connection and configuration
- `owllm/backend/utils/upload.js` - File upload and processing utilities
- `owllm/backend/utils/validation.js` - Input validation and sanitization
- `owllm/backend/db.js` - Database initialization and connection management

4.7.5. Frontend Implementation

The frontend is implemented using vanilla HTML, CSS, and JavaScript with role-based organization:

- **Student Interface** - `owllm/frontend/pages/student/` and `owllm/frontend/scripts/student/`
- **Educator Interface** - `owllm/frontend/pages/educator/` and `owllm/frontend/scripts/educator/`
- **Admin Interface** - `owllm/frontend/pages/admin/` and `owllm/frontend/scripts/admin/`
- **Shared Components** - `owllm/frontend/styles/` for CSS and common JavaScript utilities

This comprehensive database architecture provides the foundation for the RAG system while maintaining traditional relational data management for user accounts, courses, and system configuration. The stored procedures ensure secure, efficient data access while the vector search capabilities enable sophisticated AI interactions.

5. Evaluation, Limitations and Future Directions

5.1. Planned Evaluation Structure and Alignment with Use Cases

This subsection outlines the structured plan for evaluating the OWLLM system against its stated goals and use cases. As the final implementation has only recently been completed, formal testing and analysis remain ongoing. However, this section establishes a clear methodology for how the evaluation will be conducted, with references to specific user stories and their corresponding interface implementations. This roadmap will guide the upcoming assessment phase and provide a framework for documenting results.

5.1.1. Evaluation Overview

The OWLLM system will be evaluated based on how effectively it addresses the user needs identified in the earlier *User Stories* section. Each user story is tied to a specific task or interaction within the system and is implemented in a corresponding UI figure. The success of the system will be measured through the following criteria:

- **Functionality:** Does the system perform the intended tasks as defined by the user story?
- **Usability:** Is the interface intuitive and accessible for the target user group?
- **AI Quality:** Do the AI-generated responses (where applicable) align with the educational context and provide helpful guidance?
- **Consistency:** Are responses, layouts, and system behaviors uniform across different user roles and platforms?

5.1.2. User Story Alignment and Testing

Each user story will be manually tested and documented with the following evaluation results:

- Pass / Partial / Fail
- Brief notes on interface behavior
- Screenshot or visual evidence (if applicable)
- Feedback collected from test users (if applicable)

The evaluation will be organized by the same categories used in the user story section:

- Account Management (User Stories A1–A10; Figure 1, 2, 11)
- Course Access + Navigation (B1–B8; Figures 4–7)
- Content Interaction + LLM Tools (C1–C8; Figures 6–10, 18–19)
- Educator Course Management + Oversight (D1–D7; Figures 6, 7, 8, 9, 12, 18–19)
- Administrative Features (E1–E8; Figures 2, 14–16, 18)

Each of these story sets will be evaluated in a testing matrix (to be included in the appendix or supplementary materials) that cross-references the user role, the scenario being tested, the interface used, and the observed results.

5.1.3. AI Functionality and Feedback Integration

Because OWLLM includes integrated LLM interactions, a separate focus will be placed on evaluating:

- The performance of command triggers (e.g., `\note`, `\question`, `\relevance`, `\feedback`)
- The contextual appropriateness of AI responses within course materials
- The utility of the feedback system (Figure 18, 19) in refining future LLM behavior



Qualitative feedback collected through the system's built-in feedback panel will also contribute to the evaluation, offering insight into real user experience, satisfaction, and limitations not otherwise captured by technical tests.

5.1.4. Planned Deliverables

The final evaluation report will include:

- A complete matrix mapping each user story to a tested outcome
- Summaries of AI response effectiveness across educational queries
- An executive summary discussing the system's strengths, weaknesses, and observed patterns in user interaction

This planned structure ensures that the OWLLM system is evaluated not just by technical functionality but also by its educational value and alignment with its intended use cases. While the results of these tests will be incorporated once full evaluation is complete, this subsection provides a transparent and reproducible methodology for confirming that the OWLLM prototype fulfills its design objectives.

5.2. Evaluation Results

This section outlines the current structure and evaluation framework for the OWLLM system. Preliminary results are being incorporated as manual testing of user stories, AI interaction behavior, and interface functionality progresses.

5.2.1. User Story Evaluation Matrix

Table I presents a mapping of each user story to its evaluated outcome. Each entry includes the story ID, a brief scenario summary, referenced figure(s), the test result (Pass, Partial, Fail), and evaluator notes.



TABLE I
USER STORY EVALUATION MATRIX (EXCERPT)

ID	Scenario	Figure(s)	Result	Evaluator Notes
A1	Student login with email/password	Fig. 1	Passed	All functionalities confirmed
A2	Educator login with email/password	Fig. 1	Passed	All functionalities confirmed
A3	Admin login with username/password	Fig. 2	Passed	All functionalities confirmed
A4	Student access profile data	Fig. 11	Passed	All functionalities confirmed
A5	Educator access profile data	Fig. 11	Passed	All functionalities confirmed
A6	Admin access profile data	Fig. 11	Passed	All functionalities confirmed
A7	Student update email/password	Fig. 11	Passed	All functionalities confirmed
A8	Student update profile information (major, minor, interests, goals)	Fig. 11	Passed	All functionalities confirmed
A9	Educator update email/password	Fig. 11	Passed	All functionalities confirmed
A10	Admin update email/password	Fig. 11	Passed	All functionalities confirmed
B1	Student access list of enrolled courses	Fig. 4, Fig. 5	Passed	All functionalities confirmed
B2	Student select a course	Fig. 5, Fig. 6	Passed	All functionalities confirmed
B3	Student directed to main course page	Fig. 6	Passed	All functionalities confirmed
B4	Student view and select course material	Fig. 7	Passed	All functionalities confirmed
B5	Educator access list of assigned courses	Fig. 5	Passed	All functionalities confirmed
B6	Educator select a course	Fig. 5	Passed	All functionalities confirmed
B7	Educator directed to main course page	Fig. 6	Passed	All functionalities confirmed
B8	Educator view and select course material	Fig. 7	Passed	All functionalities confirmed
C1	Student select/display/download course material	Fig. 6, Fig. 7	Passed	All functionalities confirmed
C2	Student save notes/questions using \note or \question	Fig. 7	Passed	All functionalities confirmed
C3	Student ask question with LLM tool and get regulated response	Fig. 7	Partial	Most queries were properly answered, with the occasional hallucination, slightly irrelevant response, or error related to LLM settings and restraints such as max context length that cant be changed via admin settings
C4	Student view, update, and delete saved notes/questions	Fig. 8, Fig. 9	Partial	All functionalities were confirmed except for deleting questions, which is an intended design to consider the importance of questions and how they serve as a dynamic record of evolving understanding, pinpointing areas for deeper exploration.
C5	Student filter/search notes/questions by keyword or material	Fig. 8, Fig. 9	Passed	All functionalities confirmed
C6	Student use \relevance command	Fig. 7, Fig. 10	Passed	All functionalities confirmed
C7	Student use \feedback command	Fig. 18, Fig. 19	Passed	All functionalities confirmed
C8	Student view saved relevance responses	Fig. 10	Passed	All functionalities confirmed
D1	Educator upload course materials	Fig. 12	Passed	All functionalities confirmed
D2	Educator view/play/download selected course material	Fig. 6	Passed	All functionalities confirmed
D3	Educator view list of enrolled students	N/A	Passed	All functionalities confirmed
D4	Educator view student profiles, notes, and questions	Fig. 8, Fig. 9	Passed	All functionalities confirmed
D5	Educator view and respond to student questions	Fig. 9	Passed	All functionalities confirmed
D6	Educator use LLM tool to ask about course content	Fig. 7	Passed	All functionalities confirmed
D7	Educator use \feedback command	Fig. 18, Fig. 19	Passed	All functionalities confirmed
E1	Admin manage schools (add, edit, delete)	Fig. 14	Passed	All functionalities confirmed
E2	Admin manage students (add, edit, delete)	Fig. 12	Passed	All functionalities confirmed
E3	Admin manage educators (add, edit, delete)	N/A	Passed	All functionalities confirmed
E4	Admin manage courses (add, edit, delete)	N/A	Passed	All functionalities confirmed
E5	Admin manage course enrollments	Fig. 15	Passed	All functionalities confirmed
E6	Admin configure LLM settings	Fig. 16	Passed	All functionalities confirmed
E7	Admin view and analyze feedback	Fig. 18	Passed	All functionalities confirmed
E8	Admin enable/disable specific features (e.g., relevance analysis)	Fig. 16	Passed	All functionalities confirmed

5.2.2. AI Interaction and Command Trigger Testing

This subsection summarizes the evaluation structure for AI performance. It includes the parsing accuracy of commands (e.g., \note, \question, \relevance, \feedback), the relevance and correctness of responses, and engagement with the feedback system. Results are being compiled and added as testing concludes.

- **Command Parsing Accuracy:** Upwards of 95-100% of all command queries triggered their intended function. Notes and questions were properly saved, along with the dynamic functionality provided for students and educators to go back and forth for questions and answers.
- **Relevance Response Considerations:** The relevance response is inherently subjective due to individual perception of how well the LLM connects course material to career goals, research interests, and profile information. However, it is confirmed that the relevance command consistently incorporates profile information and makes a relative attempt to find connections to the respective course material. Even with seemingly unrelated topics in preliminary tests, the command was able to generate responses providing the user with ideas to consider. Further subjective evaluation is needed to determine if these responses genuinely pique a user's drive, passion, or curiosity related to their career goals and research interests.
- **Feedback System Engagement:** The feedback system and command functioned properly, saving notes and feedback as intended, which could then be viewed in the dedicated feedback section. One minor observation is that the command trigger within the LLM interface does not allow for category/type selection for feedback, unlike the modal that appears after closing the LLM interface. Based on a relatively minimal evaluation by the student working on the project and the thesis committee, approximately 8-10 feedback submissions were received. The general tone of these submissions was positive, with comments like "nice job," "quite impressive," and "good job." Some feedback also included comments on smaller bugs or verification/clarification regarding previously fixed issues.
- **General Query Response Accuracy** This section addresses the accuracy of responses to general queries posed to the assistive LLM tool, as highlighted by the user story: "As a student, I should be able to type any question into the assistive LLM tool, and have it respond with a regulated, fine-tuned response." While this user story primarily refers to the LLM's regulation and fine-tuning for an educational context, it also implies the expectation of accuracy and correctness in the generated responses. Given the open-ended nature of this user story—as any question can be asked—a definitive, static evaluation of accuracy is inherently challenging and remains an ongoing process. During the evaluation phase, responses to general queries and questions were found to be relatively accurate and correct. This held true whether the LLM was referring to the current course material or retrieving general information. A helpful feature for gauging accuracy was the LLM's ability to indicate whether a response was generated using course material, general LLM knowledge, or a mix of both, providing transparency to the user about the source of information.

5.3. Summary Analysis and Observations

The categories below summarize key evaluation themes. Final values and insights are being added as testing and review are completed.

- Strengths: [e.g., Reliable role-based navigation, intuitive UI design, strong contextual understanding by the AI.]
- Weaknesses: [e.g., Occasional parsing failures, delayed AI response for large content.]
- Opportunities for Improvement: [e.g., Enhanced handling of pedagogical tone, improvements in command robustness.]

The evaluation framework described above provides a structured basis for confirming system behavior against its design goals. Results continue to be integrated as testing progresses.

5.3.1. Strengths

The evaluation highlights several key strengths of the system, primarily revolving around its robust role-based navigation and intuitive user interface design. The role-based navigation functions seamlessly, demonstrating proper dynamics between different user types. For instance, administrators can effectively create users and assign educators and students to their respective classes. Concurrently, students can proficiently take notes and pose questions, while educators can view these notes and provide real-time answers. This real-time interaction, with only an occasional need for page reloads, significantly enhances the collaborative learning environment.

Furthermore, the user interface (UI) design has proven to be highly intuitive and easy to navigate with minimal testing. Its straightforward layout avoids unnecessary complexity, and the inclusion of tooltips, particularly for LLM settings, provides users with helpful context and understanding when hovering over components.

A significant strength lies in the AI's strong contextual understanding. When student or educator queries are well-formatted, properly spelled, and directly align with course material, the AI demonstrates excellent comprehension. It effectively fetches relevant information from the course content and can also draw upon relatable general knowledge. Crucially, the system transparently indicates whether the response was derived from course material, general knowledge, or a combination of both, providing valuable insight into the AI's reasoning.

5.3.2. Weaknesses

Despite its strengths, the evaluation has identified several areas of weakness, some of which require more intensive assessment. A notable concern revolves around the AI's performance when queries are improperly formatted or when a student asks a question that, while seemingly valid to them, doesn't align with correct or accurate answers within the system's knowledge base. A more fundamental weakness is the difficulty in definitively determining how well the tool genuinely enhances a student's understanding or learning process. While the application aims to improve course interactions and appeal to learning through an AI assistant, it's intended as a supplement, not a replacement, for traditional learning. Therefore, gaining a valid overall understanding of the tool's actual impact on student learning remains challenging.

During the evaluation phase, there were instances where the tool did not function as expected due to missing code components or other unrelated factors. For example, some file types, such as .docx, were not viewable on a Mac, unlike on a Windows computer. Similarly, while some presentation (.pptx) files could be processed, others could not, regardless of whether the issue stemmed from code errors or the document's inherent formatting. This indicates that the application is not perfectly adaptable to all scenarios and may struggle to adjust or handle external factors effectively.

Another identified weakness is the inability of the "upload YouTube video" feature to process visual content, only extracting audio. This limitation stems from the specific library used, which appears to lack the capability or legal authorization to download full video content. While inconvenient for users expecting a complete video, the audio plays fine, and its content can still be processed by the LLM, aligning with the core objective of utilizing the material for AI assistance.

The application also exhibits a significant limitation in its inability to understand diagrams, images, or other non-textual attachments within documents. Course material is primarily extracted for its text, which then forms the basis for vector embeddings and contextual understanding by the LLM. Consequently, if slides, documents, or videos contain visual elements without recognizable text, they are excluded from the processed content and, therefore, from the LLM's comprehension. While images uploaded separately can be processed by the AI's dedicated image processing tools, their integration within other materials presents a challenge.

A practical limitation is the course material upload size limit imposed by the Supabase database storage. The current free plan restricts uploads to 50MB, which is often insufficient for recorded lectures or high-quality videos that can easily exceed an hour in length. This necessitates downscaling content quality to fit within the limit. While this is a current weakness for the demo application,

upgrading to the Supabase Pro plan could significantly increase storage capacity to approximately 50GB, effectively transforming this limitation into a strength.

Finally, the monetary cost associated with operating the application is a consideration. While token-based costs for LLM vector embedding and general queries are relatively low, funds are still required for these processes. Similarly, hosting on Google Cloud Platform (GCP) also incurs costs for virtual machine usage. While these costs would need to be factored into any scaling efforts, initial calculations suggest that scalability is highly feasible, particularly when considering the financial structure of educational institutions where a small portion of tuition fees could cover the costs incurred by the LMS or AI assistant. This perspective could mitigate concerns about implementation costs for institutions.

5.3.3. Rough Estimate of Scaled Cost Calculations

I. Cost Equation

$$\text{Total Cost} = \text{Transcription Cost} + \text{Embedding Cost} + \text{Vision Cost} + \text{LLM Usage Cost}$$

5.3.3.a. A. Transcription (Whisper):

$$\text{Transcription Cost} = (L \times D_l + V \times D_v) \times C_l$$

L : Number of lectures

D_l : Duration of each lecture (minutes)

V : Number of video assignments

D_v : Duration of each video (minutes)

C_l : Cost per minute of Whisper (approx. \$0.006)

5.3.3.b. B. Embedding (Text, Slides, etc.):

$$\text{Lecture Tokens} = L \times D_l \times \text{WPM} \times T_k$$

$$\text{Textbook Tokens} = \text{Chapters} \times \text{Pages per chapter} \times \text{Words per page} \times T_k$$

$$\text{Slide Tokens} = \text{Slides} \times \text{Words per slide} \times T_k$$

$$\text{Assignment Tokens} = \text{Assignments} \times \text{Words per assignment} \times T_k$$

$$\text{Video Homework Tokens} = V \times D_v \times \text{WPM} \times T_k$$

$$\text{Total Tokens} = \text{Sum of all tokens above}$$

$$\text{Embedding Cost} = \left(\frac{\text{Total Tokens}}{1000} \right) \times C_e$$

T_k : Tokens per word ≈ 1.33

C_e : Embedding cost per 1,000 tokens $\approx \$0.0004$

5.3.3.c. C. Vision (Captured Video Frames for Context):

$$\text{Vision Cost} = S \times F \times C_f$$

S : Number of students

F : Frames captured per student

C_f : Cost per frame $\approx \$0.05$

5.3.3.d. D. LLM Usage (Queries, Feedback, Summarization, Tutoring, etc.):

$$\text{LLM Usage Cost} = Q \times T_q \times \left(\frac{C_q}{1000} \right)$$

Q : Total queries (e.g., per student $\times$ number of students)

T_q : Average tokens per query + response

C_q : Cost per 1,000 tokens (varies by model)

GPT-4o : \$0.02 per 1K tokens

GPT-3.5 : \$0.0035 per 1K tokens

II. Parameter Sets

5.3.3.e. Less Intensive Course

- $L = 20$, $D_l = 30$ minutes, WPM = 140
- Chapters = 8, Pages per chapter = 8, Words per page = 400
- Slides = 400, Words per slide = 15
- Assignments = 5, Words per assignment = 600
- $V = 5$, $D_v = 5$ minutes
- $S = 30$, $F = 15$
- $T_k = 1.33$
- $Q = 100 \times 30 = 3000$
- $T_q = 700$
- Cost constants: $C_l = 0.006$, $C_e = 0.0004$, $C_f = 0.05$, $C_q = 0.02$

5.3.3.f. More Intensive Course

- $L = 50$, $D_l = 60$ minutes, WPM = 160
- Chapters = 20, Pages per chapter = 15, Words per page = 600
- Slides = 2500, Words per slide = 25
- Assignments = 15, Words per assignment = 1500
- $V = 15$, $D_v = 15$ minutes
- $S = 100$, $F = 30$
- $T_k = 1.33$
- $Q = 300 \times 100 = 30000$
- $T_q = 1000$
- Cost constants: same as above

III. Final Cost Summary (GPT-4o)

Component	Less Intensive	More Intensive
Whisper Cost	\$3.90	\$20.70
Embedding Cost	\$0.07	\$0.42
Vision Frame Cost	\$22.50	\$150.00
LLM Usage Cost	\$42.00	\$600.00
Total Cost	\$68.47	\$771.12

Justification and Implications

The development of OWLLM, along with the limited testing phase evaluation, resulted in only a \$1.09 deduction from the total available credits of \$20. This reflects a minimal resource footprint during early-stage development and demonstrates responsible and efficient use of OpenAI resources.

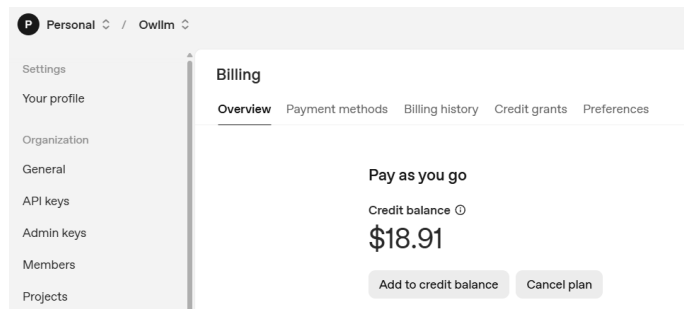


Fig. 23. Credit usage overview during OWLLM development and testing

Moreover, the cost modeling shows that, at scale, a full course implementation using OpenAI technologies could range from \$0 to \$1000—an amount modest compared to the typical \$1000–\$5000 that students pay per course. When distributed across multiple students and sessions, this cost becomes a relatively small investment that could be factored into tuition or institutional budgets. Such spending would contribute toward a technically enhanced LMS experience supported by AI, with the persistent overall arching goal to be to improve course engagement and educational outcomes.

5.3.4. Opportunities for Improvement

Several opportunities exist to enhance the system, many of which directly address the weaknesses outlined above. Reconsidering and reiterating these identified weaknesses serves as an excellent starting point for outlining areas of improvement.

Beyond these, there are opportunities to refine the visual interface and styling. This includes addressing minor bugs such as partially visible drop-down menus, odd spacing, or components that appear too closely packed within the application. Addressing these aesthetic and functional inconsistencies would contribute to a more polished and user-friendly experience.

Furthermore, a dedicated section within the document will delve into potential future directions and expanded functionalities. This section will provide a more detailed exploration of additional areas for improvement and potential enhancements that could further elevate the system's capabilities.

5.4. Limitations

As has been previously mentioned, this project comes with several limitations. Ideally, a true embedded integration with an existing learning management system (LMS), such as Blackboard—which serves millions of users—would be desirable. However, this is not feasible due to the proprietary nature of such platforms. Individual students typically do not have access to the source code, nor the permissions required to modify or extend the system's functionality.

Blackboard's proprietary architecture seems to restrict direct modification or embedding of external tools like OWLLM without official collaboration and access to backend systems. However, it does seem that Blackboard can support integration through industry standards such as Learning Tools Interoperability (LTI) and RESTful APIs. After some consideration, developing an LTI-compliant version of OWLLM or leveraging Blackboard's APIs could facilitate further integration. Yet, these approaches would require more extensive coordination with institutional administrators, admin-level permissions, and the use of third-party developer frameworks—all of which introduce additional complexity and limitations. Despite these barriers, this remains a promising direction for future work and is worth further investigation, especially considering that Blackboard is currently used at SCSU.

Given this constraint, the current project is intended to serve as a proof of concept. The goal is to demonstrate the potential value of a supplemental AI-driven tool that, if developed and

integrated at scale, could enhance the learning experience for students and educators within an LMS environment. These limitations also shape the project's scope, focusing on simple core features rather than more advanced capabilities involving full AI and large language model (LLM) integration.

There are also challenges from the more difficult, stubborn student perspectives that might exist. One of the project's aims is to address common misuses of LLM technology in academic contexts. However, it cannot fully prevent students from accessing external tools beyond the LMS. While the tool can be fine-tuned to encourage proper use—through tailored responses and restrictions such as disabling copy-paste functionality—its effectiveness ultimately depends on how students choose to engage with it. Proximity to course materials does not guarantee optimal or ethical usage, and the tool can only suggest appropriate learning behavior rather than enforce it.

Additional limitations include the current scope of supported file formats and processing capabilities. While the system supports common educational file types (PDF, DOCX, PPTX, MP4, MP3, etc.), it may not handle all specialized academic content formats or complex multimedia presentations. The RAG system's effectiveness is also limited by the quality and structure of uploaded content, requiring well-formatted documents for optimal vector embedding generation.

5.5. Proposed User Study Design

Despite the limitations, constraints, and weaknesses identified above as the results of the smaller-scale evaluation of the application, here is an outlined suggested longitudinal experiment that can better evaluate the multifaceted impact of LLM tools on student learning, extending beyond mere academic performance to encompass cognitive, emotional, and behavioral dimensions. This proposed study would significantly enhance the depth of both qualitative and quantitative data collection, moving beyond traditional academic metrics like GPA, test scores, and assignment completion rates.

To gain a more comprehensive understanding of the cognitive engagement, the refined design would incorporate precise tracking of time-on-task when students interact with LLM tools and their overall dedicated study time, correlating these efforts with measured learning outcomes. This could involve logging tool usage data and requiring students to maintain detailed self-reported study logs. Furthermore, pre- and post-intervention assessments would be designed to measure specific learning gains, potentially utilizing standardized concept inventories or complex problem-solving tasks that necessitate deeper understanding. Qualitative data, gathered through interviews and open-ended survey questions, would also explore the students' perceived cognitive load when using LLM tools compared to traditional study methods, providing crucial insights into their mental effort.

Understanding the emotional experience of learners is equally vital. The proposed study would employ regular sentiment analysis of open-ended survey responses and detailed interview transcripts to identify recurring patterns of frustration, satisfaction, confidence, or anxiety directly related to LLM tool usage. We would also incorporate validated psychological scales into surveys to measure changes in student motivation towards learning and their self-efficacy, specifically their belief in their ability to succeed with and without the assistance of LLM tools.

Regarding behavioral changes, the study would aim to assess the influence of LLM tools on procrastination reduction through both self-reported measures and potentially objective data like submission timestamps relative to deadlines. In-depth qualitative data would explore how LLM tools integrate into students' existing study habits, whether they encourage more active learning strategies or passive consumption of information, and any shifts in their help-seeking behaviors. A thorough analysis of LLM conversation data would also be conducted to understand patterns of usage, the types of queries students pose, and the evolution of their interaction styles with the tools over time.

To achieve this depth, a dedicated user study component would be embedded within the longitudinal design. This would involve a subset of participants engaging in structured learning tasks while actively using LLM tools. During these sessions, think-aloud protocols would be

employed, where participants verbalize their thoughts and strategies, providing real-time insights into their cognitive processes. Immediately following task completion, in-depth post-task interviews would probe specific interaction points, emotional responses, and perceived utility. For an even more comprehensive understanding, future iterations of this research could explore non-invasive physiological measures like eye-tracking to assess attention and engagement, or galvanic skin response for emotional arousal, during controlled LLM interaction sessions, though this would necessitate specialized equipment and expertise. To capture the evolving user experience throughout the semester, more frequent and structured qualitative data collection points, such as bi-weekly focus groups and regular journaling prompts, would be implemented.

Naturally, all ethical safeguards, including explicit informed consent, meticulous data anonymization, and secure storage, would be rigorously maintained throughout the study. While potential limitations, such as inherent biases in self-reported data and the influence of external life factors on academic performance, would still be acknowledged, the proposed enhanced data collection methods aim to triangulate findings and provide a more robust understanding of the complex interplay between LLM tools and the learner's cognitive, emotional, and behavioral landscape. Ultimately, this refined approach would yield invaluable insights into not just the efficacy of LLM tools in improving academic performance, but crucially, how they shape the student's learning journey, fostering a deeper understanding of personalized AI-enhanced education.

5.6. Future Directions and Extended Functionality

5.6.1. Additional Features for Enhanced Educational Experience

As a foundational proof-of-concept, the OWLLM system lays the groundwork for a wide range of advanced features that could significantly deepen the educational value it offers. One major area of expansion is the development of advanced analytics and learning insights. These would include comprehensive dashboards for tracking student engagement patterns, visualizing content interaction metrics, and monitoring progress over time. Such insights could support educators in identifying struggling students or underutilized materials.

Another future direction involves the implementation of personalized learning paths. Leveraging AI models, the system could offer tailored recommendations for additional course materials, curated readings, and study strategies based on a student's academic performance, learning preferences, and declared interests. Alongside this personalization, OWLLM could incorporate collaborative learning features, such as group chat spaces tied to specific courses, shared note-taking environments, and peer-to-peer question forums. These tools would facilitate both academic support and community building.

The system could also evolve to offer more sophisticated multimodal content processing capabilities. This would include automated transcription of video and audio content, extraction of key discussion points, and timestamp-based navigation for quickly locating relevant sections of multimedia resources. Coupled with this, OWLLM could support real-time assessment tools such as in-line quizzes, adaptive knowledge checks, and immediate feedback mechanisms—features that enhance interactivity and allow learners to evaluate their understanding as they progress through course materials.

5.6.2. Extended Educator Functionality and Use Cases

Educators stand to benefit greatly from a deeper integration of AI within the OWLLM platform. One important avenue for future development is content creation assistance. This would include tools for generating lecture summaries, drafting practice questions, and producing supplementary materials derived from existing course content. Such functionality would streamline course design and reduce cognitive load for educators.

Moreover, the system could offer detailed student performance analytics, enabling instructors to track both individual and class-wide trends in engagement and learning. These insights would help identify knowledge gaps and allow for timely interventions. OWLLM could also support automated grading workflows by evaluating written assignments, generating formative feedback,

and highlighting common student misconceptions.

Curriculum optimization is another promising direction, where the AI system would analyze course structures and recommend improvements to content sequencing, topic emphasis, or resource allocation. Combined with personalized student support tools, this would enable educators to provide data-driven academic guidance tailored to individual learner needs. Finally, the system could assist in bridging the gap between instruction and research by suggesting connections between course topics and current academic publications, enabling educators to enrich classroom discussions with up-to-date scholarship and real-world applications.

5.6.3. Advanced AI Integration and Capabilities

As AI models continue to evolve, OWLLM is well positioned to incorporate more advanced capabilities that support multi-modal learning and persistent intelligent assistance. Future versions of the platform could integrate multi-modal AI models capable of processing visual and auditory content. These would allow the system to understand and contextualize diagrams, recorded lectures, and spoken explanations with greater semantic depth.

Another enhancement could be the introduction of conversational AI agents—persistent digital tutors that maintain context across multiple interactions. These agents could serve as long-term academic companions, tracking a student's learning trajectory and adapting their support accordingly. Underpinning these features would be adaptive learning algorithms that continuously refine their recommendations and responses based on user behavior, feedback, and performance data.

Further improvements could involve knowledge graph integration, allowing the system to link concepts across courses and disciplines. This would help students develop cross-curricular understanding and explore broader connections within their fields of study. Additionally, OWLLM could employ natural language generation technologies to automatically produce summaries, explanatory content, and even customized study guides, greatly expanding its role as a generative educational assistant.

5.6.4. General Applications and Broader Impact

Although OWLLM is designed primarily for academic learning environments, the underlying architecture and educational principles can be extended to serve a variety of other domains. In corporate training and professional development, the retrieval-augmented generation (RAG) approach can be adapted to support workplace learning programs, certification preparation, and skills acquisition across industries. Similarly, OWLLM could find application in healthcare education, where it could help clinicians and trainees engage with complex clinical documentation, procedural guidelines, and evidence-based medical knowledge.

In research and academic publishing, OWLLM's AI-driven document processing and synthesis tools could support literature review, citation organization, and scholarly writing. Moreover, the system's potential for accessibility and inclusive education is significant. AI-driven content adaptation features—such as automated language translation, text simplification, and multi-modal summarization—could help learners with diverse needs better access educational content. Finally, the platform could support lifelong learning initiatives by providing accessible, adaptive educational experiences to adult learners seeking continuing education, personal enrichment, or professional growth.

5.6.5. Enhancing Equity and Accessibility through Inclusive AI Design

Building on the identified potential for accessibility and inclusive education, future enhancements to the OWLLM system could specifically target the design considerations necessary to support neuro-divergent students and English Language Learners (ELLs). This proactive approach would help to address the issue of educational tools potentially disproportionately benefiting already advantaged students, fostering a truly equitable learning environment for all users.

For neuro-divergent learners, incorporating different interface pacing modes would allow stu-

dents granular control over the speed at which AI-generated content is displayed and responses are provided. This would significantly reduce cognitive load and enhance comprehension by allowing self-regulated processing of information. Furthermore, offering simplified response options would present complex information in more digestible, direct formats, minimizing ambiguity and supporting clear understanding for learners who benefit from concise communication. While the current system primarily processes text, the future integration of multimodal AI models could be specifically leveraged to generate visual summaries of key concepts from course materials, providing an alternative and accessible pathway to understanding for visual learners.

For English Language Learners (ELLs), refining the AI's natural language generation capabilities to provide output in multiple reading levels would enable students to select the complexity of explanations, making challenging academic texts more accessible as their proficiency develops. Additionally, integrating a feature to automatically add definitions for complex vocabulary directly within AI responses or as interactive elements alongside course content would provide immediate linguistic support, aiding both comprehension and vocabulary acquisition. While automated language translation is already recognized as a valuable future feature, these specific linguistic supports would ensure the AI effectively serves as a comprehensive language learning aid, bridging gaps in understanding.

Implementing these targeted design considerations aligns with the broader goal of making OWLLM an adaptive and responsive learning companion, ensuring it provides truly inclusive and equitable support that enhances learning outcomes across diverse student populations. This approach reinforces the system's commitment to addressing "learner affect and emotional diversity" by recognizing and adapting to a wider spectrum of cognitive and linguistic needs.

5.6.6. Technical Enhancements and Scalability

To support the demands of broader deployment and future features, OWLLM must undergo several technical enhancements. One priority is the development of a distributed architecture that can scale to accommodate institutions of varying sizes, including universities with multiple campuses and thousands of users. This would ensure reliable system performance across diverse usage environments.

In addition, the platform could be augmented with offline capabilities, allowing it to function in regions with limited or unstable internet connectivity. This would make OWLLM more suitable for global deployment, particularly in underserved areas. Security and compliance features would also need to be improved. Future versions could implement stronger encryption protocols, user auditing tools, and adherence to data privacy regulations specific to educational institutions.

To foster integration with existing academic systems, OWLLM could support an open API ecosystem that allows for third-party tool and platform interoperability. Performance optimization would also be critical, particularly for maintaining fast and reliable AI responses, efficient file processing, and consistent user experience as the system scales.

5.6.7. Research and Evaluation Opportunities

Beyond feature development, OWLLM opens opportunities for continued research in educational technology and human-computer interaction. One area of interest is learning analytics research, where longitudinal data from user interactions can be analyzed to understand patterns of engagement, student success, and instructional effectiveness.

The system could also support studies investigating the impact of AI-assisted tools on student performance, satisfaction, and retention. Comparative effectiveness research could be conducted to assess the relative utility of different AI models, vector embedding strategies, or interaction modalities used in OWLLM.

Moreover, the platform provides a foundation for longitudinal impact assessments examining how sustained use of AI tools influences academic outcomes and career trajectories. These research opportunities not only validate the system's effectiveness but also contribute to the broader academic understanding of AI's role in transforming education.

In summary, the OWLLM prototype establishes a modular and scalable framework that is well suited for continued development. The directions outlined above highlight both the pedagogical and technical avenues through which OWLLM could evolve into a fully featured, research-backed educational assistant platform.

Citations, Sources and References

- [1] D. Litman, "Natural language processing for enhancing teaching and learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016.
- [2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.
- [3] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Draft, 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [4] M. Boden, *Artificial Intelligence: A Very Short Introduction*, Oxford University Press, 2018.
- [5] J. H. Korteling, G. C. van de Boer-Visschedijk, R. A. Blankendaal, R. C. Boonekamp, and A. R. Eikelboom, "Human-versus artificial intelligence," *Frontiers in Artificial Intelligence*, vol. 4, p. 622364, 2021.
- [6] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: An overview," *J. Phys.: Conf. Ser.*, vol. 1142, p. 012012, 2018.
- [7] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. New York, NY: Basic Books, 2015.
- [8] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, UK: Cambridge Univ. Press, 2014.
- [9] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.
- [10] T. M. Mitchell, *Machine Learning*. New York, NY: McGraw-Hill, 1997.
- [11] Y. Wang, J. J. Vogel, and W. Ran, "The impact of AI on student learning outcomes in higher education: A systematic review," *J. Educ. Technol. Soc.*, vol. 23, no. 2, pp. 1–11, 2020.
- [12] W. Holmes, M. Bialik, and C. Fadel, *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning*, Boston, MA: Center for Curriculum Redesign, 2019.
- [13] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education – where are the educators?" *Int. J. Educ. Technol. High. Educ.*, vol. 16, p. 39, 2019.
- [14] C. Luckin et al., "Intelligence unleashed: An argument for AI in education," Pearson Education, 2016. [Online]. Available: <https://www.pearson.com>
- [15] D. Litman and S. Silliman, "ITSPOKE: An intelligent tutoring spoken dialogue system," in *Demonstration Papers at HLT-NAACL 2004*, 2004, pp. 5–8.
- [16] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies, and why do we need them?" *IEEE Intell. Syst.*, vol. 14, no. 1, pp. 20–26, Jan. 1999.
- [17] D. Becker, K. Hu, and M. Xu, "Automated feedback generation for student writing using transformer-based language models," in *Proc. 13th Int. Conf. Educ. Data Mining (EDM)*, 2020, pp. 630–633.
- [18] G. Khensous, K. Labed, and Z. Labed, "Exploring the evolution and applications of natural language processing in education," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 3713–3738, 2023.
- [19] R. Kumar and R. S. Thakur, "Recommender systems in e-learning: A review," *Int. J. Comput. Appl.*, vol. 150, no. 5, pp. 25–32, 2016.

- [20] A. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: A review of state-of-the-art and research challenges," *Artif. Intell. Rev.*, vol. 49, no. 4, pp. 1–42, 2018.
- [21] D. Winkler and B. Söllner, "Unleashing the potential of chatbots in education: A state-of-the-art analysis," in *Proc. 2018 IEEE Global Engineering Education Conf. (EDUCON)*, pp. 1704–1713.
- [22] C. F. Lee, "Artificial intelligence in education: Challenges and opportunities for sustainable development," *Sustainability*, vol. 14, no. 1, p. 141, 2022.
- [23] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, 2023.
- [24] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009.
- [25] J. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [26] Y. Goldberg, "A primer on neural network models for natural language processing," *J. Artif. Intell. Res.*, vol. 57, pp. 345–420, 2016.
- [27] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [28] M. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 15–21, Mar.–Apr. 2013.
- [29] M. D. Sherin, "A balancing act: Developing a discourse analysis for mathematics classrooms," *J. Learn. Sci.*, vol. 10, no. 4, pp. 477–522, 2001.
- [30] M. T. Mohamad and S. Tasir, "Educational data mining: A review," *Procedia Soc. Behav. Sci.*, vol. 97, pp. 320–324, 2013.
- [31] A. Graesser, X. Hu, B. Nye, and Y. Liu, "AutoTutor: A learning companion for improving deep learning," *J. Educ. Psychol.*, vol. 106, no. 4, pp. 900–908, 2014.
- [32] M. Woolf, W. Burleson, T. Arroyo, D. Dragon, and S. Cooper, "Affect-aware tutors: Recognizing and responding to student affect," *Int. J. Learn. Technol.*, vol. 4, no. 3/4, pp. 129–164, 2009.
- [33] R. Pekrun and L. Linnenbrink-Garcia, *International Handbook of Emotions in Education*. New York, NY: Routledge, 2014.
- [34] M. von Davier and V. Yaneva (Eds.), *Advancing Natural Language Processing in Educational Assessment*, 1st ed. New York, NY: Routledge, 2023.
- [35] A. Graesser, S. D'Mello, and B. Cade, "Instruction based on tutoring," in *Handbook of Research on Learning and Instruction*, 2nd ed., R. E. Mayer and P. A. Alexander, Eds. New York, NY: Routledge, 2016, pp. 400–421.
- [36] X. Chen, D. Zou, H. Xie, G. Cheng, and C. Liu, "Two decades of artificial intelligence in education: Contributors, collaborations, research topics, challenges, and future directions," *Educ. Technol. Soc.*, vol. 25, no. 1, pp. 28–47, 2022.
- [37] V. Yaneva, C. Zhai, and M. von Davier, "Challenges in NLP-based educational assessment: Generalizability and fairness," in *Advancing Natural Language Processing in Educational Assessment*, V. Yaneva and M. von Davier, Eds. New York, NY: Routledge, 2023, ch. 7.
- [38] X. Chen, D. Zou, H. Xie, G. Cheng, and C. Liu, "Two decades of artificial intelligence in education: Contributors, collaborations, research topics, challenges, and future directions," *Educ. Technol. Soc.*, vol. 25, no. 1, pp. 28–47, 2022.
- [39] R. Calvo and S. D'Mello, "Affect detection: An interdisciplinary review of models, methods,

- and their applications,” *IEEE Trans. Affect. Comput.*, vol. 1, no. 1, pp. 18–37, Jan. 2010.
- [40] C. Conati and H. Maclaren, “Empirically building and evaluating a probabilistic model of user affect,” *User Model. User-Adapt. Interact.*, vol. 19, no. 3, pp. 267–303, 2009.
- [41] S. D’Mello and A. Graesser, “Monitoring affective trajectories during complex learning,” in *Affective Computing and Intelligent Interaction*, Springer, 2010, pp. 128–139.
- [42] S. Baker, K. Gowda, M. Wixon, and A. Corbett, “Towards sensor-free affect detection in educational games,” in *Proc. 2012 Int. Conf. User Model., Adaptation, and Personalization (UMAP)*, pp. 126–138.
- [43] M. von Davier and V. Yaneva (Eds.), *Advancing Natural Language Processing in Educational Assessment*, 1st ed. New York, NY: Routledge, 2023.
- [44] D. S. McNamara, “iStart: A reading strategy tutor that teaches self-explanation,” in *Handbook of Research on Educational Communications and Technology*, 4th ed., J. M. Spector et al., Eds. Springer, 2014, pp. 271–283.
- [45] V. Aleven, B. McLaughlin, R. Glenn, and K. Koedinger, “Instruction based on adaptive learning technologies,” in *Handbook of Research on Learning and Instruction*, 2nd ed., R. E. Mayer and P. A. Alexander, Eds. New York, NY: Routledge, 2016, pp. 522–560.
- [46] B. Woolf, *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning*. San Francisco, CA: Morgan Kaufmann, 2010.
- [47] S. Heffernan and N. Heffernan, “The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching,” *Int. J. Artif. Intell. Educ.*, vol. 24, no. 4, pp. 470–497, Dec. 2014.
- [48] J. Clarke-Midura and C. Dede, “Assessment, technology, and change,” *J. Res. Technol. Educ.*, vol. 45, no. 1, pp. 3–16, 2012.
- [49] D. S. McNamara, J. E. Crossley, and P. M. McCarthy, *Linguistic Features of Writing Quality*. Cambridge University Press, 2010.
- [50] V. Yaneva, C. Zhai, and M. von Davier, “Challenges in NLP-based educational assessment: Generalizability and fairness,” in *Advancing Natural Language Processing in Educational Assessment*, V. Yaneva and M. von Davier, Eds. New York, NY: Routledge, 2023, ch. 7.
- [51] R. Binns, “Fairness in machine learning: Lessons from political philosophy,” *Proc. 2018 Conf. Fairness, Accountability, and Transparency (FAT)*, pp. 149–159, 2018.
- [52] R. Binns, M. Veale, M. Van Kleek, and N. Shadbolt, “‘It’s reducing a human being to a percentage’: Perceptions of justice in algorithmic decisions,” *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst.*, 2018.
- [53] N. Selwyn, *Should Robots Replace Teachers? AI and the Future of Education*, Cambridge, UK: Polity Press, 2019.
- [54] L. Floridi and M. Chiriatti, “GPT-3: Its nature, scope, limits, and consequences,” *Minds Mach.*, vol. 30, pp. 681–694, 2020.
- [55] M. Selwyn, “Should robots replace teachers? AI and the future of education,” *Brit. J. Educ. Stud.*, vol. 68, no. 1, pp. 1–17, 2020.
- [56] E. Kasneci et al., “ChatGPT for good? On opportunities and challenges of large language models for education,” *Learn. Individ. Differ.*, vol. 103, p. 102274, 2023.
- [57] S. B. T. Kovanović and G. Siemens, “From reflection to action: Towards a framework for educationally aligned learning analytics,” *Proc. 6th Int. Conf. Learn. Analytics Knowl.*, pp. 230–239, 2016.
- [58] H. Luckin et al., *“Intelligence unleashed: An argument for AI in education,”* Pearson

Education, 2016.

- [59] P. J. Guo, "When AI tutors fail: Towards explainable and resilient student-AI partnerships," *ACM Trans. Comput.-Hum. Interact.*, vol. 30, no. 1, pp. 1–29, 2023.
- [60] N. Rane, "Enhancing the quality of teaching and learning through ChatGPT and similar large language models: Challenges, future prospects, and ethical considerations in education," *Future Prospects, and Ethical Considerations in Education*, Sep. 15, 2023.
- [61] A. J. D'Mello, A. Graesser, and B. Scherer, "AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back," in *Proceedings of the 4th International Conference on Intelligent Tutoring Systems*, 2008, pp. 61–70.
- [62] Deci, E. L., & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior*. Springer Science & Business Media.
- [63] VanLehn, K. (2011). The goal of learning is not the goal of instruction, or: What to do about the ZPD. In *Cognitive science approaches to instructional design* (pp. 1-28). Springer.
- [64] Piech, C., et al. (2015). Deep knowledge tracing. In *Advances in Neural Information Processing Systems* (pp. 505-513).
- [65] Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257-285.
- [66] Flavell, J. H. (1979). Metacognition and cognitive monitoring. *American Psychologist*, 34(10), 906-911.
- [67] Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2), 145-182.
- [68] Milner, A. D., & Goodale, M. A. (2008). *The visual brain in action* (2nd ed.). Oxford University Press.
- [69] Grill-Spector, K., & Malach, R. (2004). The human visual cortex. *Annual Review of Neuroscience*, 27, 649-677.
- [70] Posner, M. I., & Petersen, S. E. (1990). The attention system of the human brain. *Annual Review of Neuroscience*, 13, 25-42.
- [71] Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24, 167-202.
- [72] Friederici, A. D. (2011). The brain basis of language processing: From structure to function. *Physiological Reviews*, 91(4), 1357-1392.
- [73] Price, C. J. (2012). A review and synthesis of the first 20 years of PET and fMRI studies of heard speech, spoken language and reading. *NeuroImage*, 62(2), 816-847.
- [74] Hagoort, P. (2005). On Broca, brain, and binding: A new framework. *Trends in Cognitive Sciences*, 9(9), 416-423.
- [75] Fedorenko, E., & Thompson-Schill, S. L. (2014). Reworking the language network. *Trends in Cognitive Sciences*, 18(3), 120-126.
- [76] Hickok, G., & Poeppel, D. (2007). The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8(5), 393-402.
- [77] Binder, J. R., et al. (2009). Semantic processing in the brain. *Journal of Cognitive Neuroscience*, 21(6), 1111-1127.
- [78] Seghier, M. L. (2013). The angular gyrus: Multiple functions and multiple subdivisions. *The Neuroscientist*, 19(1), 43-61.
- [79] Binder, J. R., & Desai, R. H. (2011). The neurobiology of semantic memory. *Trends in*

Cognitive Sciences, 15(11), 527-536.

- [80] Craik, F. I. M., & Tulving, E. (1975). Depth of processing and the retention of words. *Journal of Experimental Psychology: General*, 104(3), 268-294.
- [81] McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3), 419-457.
- [82] Squire, L. R., & Zola-Morgan, J. T. (2011). The cognitive neuroscience of human memory since H.M. *Annual Review of Neuroscience*, 34, 259-288.
- [83] Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review*, 108(3), 624-652.
- [84] Gehring, W. J., Goss, B., Coles, M. G., Meyer, D. E., & Donchin, E. (1993). A neural system for error detection and compensation. *Psychological Science*, 4(6), 385-390.
- [85] Holroyd, C. B., & Coles, M. G. H. (2002). The neural basis of human error processing. *Psychological Review*, 109(4), 679-709.
- [86] Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275(5306), 1593-1599.
- [87] Doya, K. (2008). Modulators of decision making: Dopamine, serotonin, and noradrenaline. *Neuron*, 36(4), 725-741.