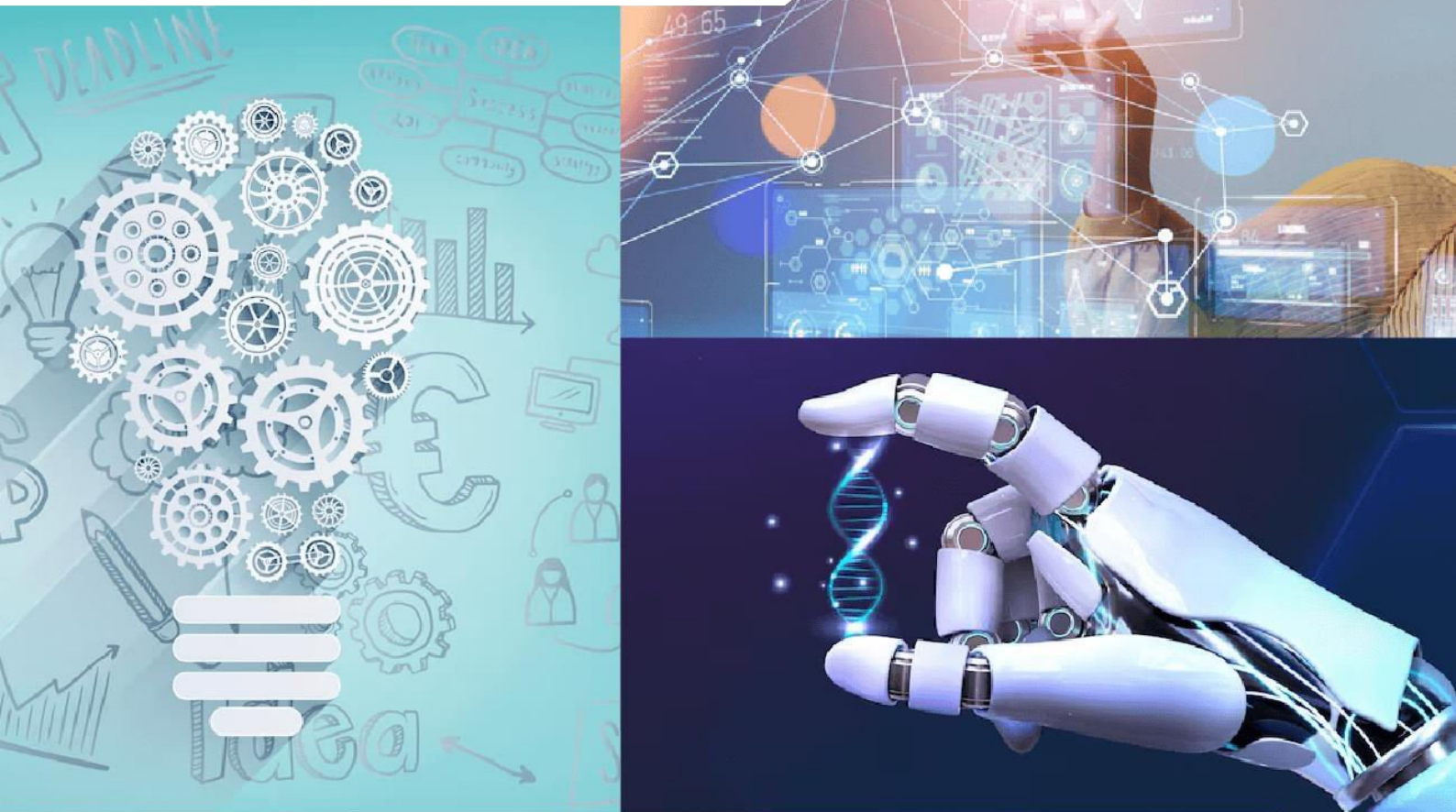




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 2, March - April 2026

Impact Factor: 8.152



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



SmartLearn: An AI Powered SmartLearn and Assessment System

Sushmitha

Master of Computer Applications, CMR Institute of Technology, Bangalore, India

ABSTRACT: Most online learning platforms fail because they rely on passive reading without checking if students actually grasp the material. This project introduces SmartLearn, a custom web application built to solve that problem by turning regular documents into interactive study sessions. To accurately evaluate what a student knows, SmartLearn runs several AI modules at the same time. First, it automatically reads uploaded syllabus files to generate short notes, quizzes, and coding tests. Second, it uses natural language processing to grade written paragraph answers based on their actual meaning rather than basic keyword matching. Finally, it features an AI voice examiner that conducts live, spoken oral tests through the microphone, acting just like a human teacher. By combining these smart testing methods with gamified leaderboards and reward points, SmartLearn transforms standard, boring online tests into a highly engaging tool that actively motivates students to succeed.

KEYWORDS: Artificial Intelligence, E-Learning Systems, Natural Language Processing, Automated Question Generation, Speech Emotion Recognition, Generative AI, Intelligent Tutoring Systems, Gamification.

I. INTRODUCTION

Software in schools is not just for holding files anymore. Computer science has reached a point where systems can actually monitor how someone studies and interact with them in real time. A big piece of this new wave involves using artificial intelligence to automatically grade assignments and figure out exactly what a student is struggling with. Getting this right is completely necessary for building better digital classrooms and virtual tutors [1], [3].

Normally, online tests are pretty basic. They rely heavily on multiple-choice questions or they force students to type in exact, word-for-word answers. If a student understands the concept but uses a different word, they still fail. Instead of getting stuck on exact spellings, newer systems try to read the actual meaning behind a sentence. Also, instead of making teachers write all the study guides by hand, modern AI can scan huge textbooks and pull out the important topics on its own [4], [9], [17].

Written tests only show one side of the picture. Listening to how a student explains a concept out loud is just as important, especially for catching quick thinking and communication skills. Right now, the tech world relies on complex transformer models to handle voice data. These models are much better than old voice recorders because they can understand context or even figure out what a student means when they stutter [13], [18], [19].

Even with all this proof showing that active involvement is better, most college management systems completely ignore these tools. They usually just track if a student clicked a video link or opened a document. Because real learning requires constant feedback and interaction, just giving out reading assignments makes these apps feel outdated [12], [20]. To fix this issue, this paper introduces SmartLearn. It is a custom web project that merges automatic study guide creation with multiple layers of testing.

What makes this project truly unique compared to other education platforms is its 'Multimodal Cognitive Assessment' approach. It does not just test a student's memory; it explicitly tests their ability to express ideas. First, it completely eliminates the need for human teachers to draw up study grids by instantly building lessons and tests straight from raw PDFs [6]. Second, it runs a proprietary 'Assessment Trifecta': the student must type out the theory, program the logic in a live coding sandbox, and finally, successfully explain the concept out loud in a spoken interview against a digital AI teacher [14]. This forces the student to prove their knowledge three completely different ways. Finally, tying all three of these advanced grading metrics into a single, aggressively gamified leaderboard creates a highly competitive environment that pushes students to actually want to study [10].

II. LITERATURE REVIEW

In the past, digital learning basically meant putting a textbook on a website. This approach caused a lot of problems because every student learns at a different speed. Because school subjects can be incredibly complex, these basic digital libraries failed to offer the kind of interaction students actually need to stay focused [9], [12]. Once deep neural networks came onto the scene, things changed. Computers started figuring out how to read and summarize syllabus documents automatically, which saved human teachers a massive amount of planning time [3], [17].

When it comes to grading, research has heavily moved towards fixing the "exact match" problem. Today, developers use transformer-based setups like BERT to get rid of old-school keyword grading scripts. Because these setups look at every word in a sentence at the same time, the computer can tell the difference between an answer that is completely wrong and one that is right but phrased a bit weirdly [14], [16], [18]. At the same time, a lot of work has gone into making AI reliably create quiz questions straight from the course documents, so that the tests match exactly what the students just read [2], [5], [7], [8].

Lately, the biggest trend in smart tutoring is mixing different types of learning together—like reading, typing, and speaking all at once [11]. Grouping these tests together makes grading much fairer. For example, if a student is bad at writing long essays, their performance in the coding practice or the oral interview might prove they actually know the material. Using a mix of tests leads to a much more accurate final grade and avoids the mistakes that happen when you only test one way [1], [6].

Educational researchers agree with this hands-on approach. If you just force a student to read for hours, they get exhausted and stop paying attention. Adding constant, low-pressure quizzes mixed with game elements like points and digital rewards directly pushes the student to keep going. Since keeping a student motivated is the hardest part of teaching, building these feedback loops into the software is critical for academic success [10], [15].

III. METHODOLOGY

To put SmartLearn together, the actual code wires up a document reader and multiple testing tools into one single platform. This is where the 'Assessment Trifecta' comes to life. The technical side relies on a few totally different tools running simultaneously: natural language processing to read the PDFs, semantic models to grade written essays, compilers to check live programming code, and audio logic to run verbal interviews. Pulling tests from all these different angles drops the grading error rate close to zero. It makes the software undeniably better at figuring out true skill levels than older apps that only rely on outdated, multiple-choice clicking.

3.1 Intelligent Document Processing and "Smart Notes" Generation

When a teacher or student drops a messy syllabus file into the app, the backend uses Generative AI to sort it out. Instead of making a human sit down and summarize a fifty-page chapter, the algorithms read the whole file to find out how the topics link up. The system automatically pulls out the main ideas, definitions, and important facts to create clean, readable "Smart Notes." Students use these notes a lot because they get an instant, focused study guide without having to ask a professor for help [5], [17].

3.2 Automated Quizzes and Semantic Evaluation Engine

Right after a user finishes reading a module, the system immediately checks if they actually paid attention. The code builds a unique set of quiz questions strictly based on the Smart Notes that were just generated [6], [7]. For the short answer text boxes, the backend runs modern transformer setups. Instead of being like an old testing site that fails you if you miss a specific keyword, SmartLearn calculates the math distance between the true concept and whatever the student typed to grade the actual idea [14], [16].

3.3 Code Execution Sandbox

Reading and writing are fine, but computer science students mostly need to write actual logic. SmartLearn includes a live coding screen directly inside the web browser. If someone types out a chunk of Python or Java, the browser sends it to an isolated backend server. This setup instantly runs the code and checks for errors, letting the AI jump in with highlighted text explaining exactly what line broke and why.

3.4 AI Voice Examiner

To stop students from just memorizing flashcards, the platform runs a live spoken interview that feels like a real college viva. When a student talks into their microphone, a Speech-to-Text model quickly turns the recorded audio into text

[20]. The AI then reads that text to check how well the student understands the technical topic. After grading the answer, the system uses Text-to-Speech to act as a virtual interviewer, asking follow-up questions to keep the conversation going [11], [19].

3.5 System Architecture

The core structure of SmartLearn is broken into separate, isolated pieces so the website does not freeze while thinking.

The Presentation Layer: Built with React, this handles the main visuals. It runs the screens where users read notes, type code, and look at their score charts.

The Application Layer: Running on server tools like Node.js and Python Flask, this part safely stores user passwords, handles file uploads, and directs web traffic.

The AI Processing Microservice: A separate Python server specifically set up to do the heavy mathematical lifting for the AI reading tasks, text grading, and microphone audio conversions [18].

The Database Layer: This keeps track of everything long-term, storing user profiles, textbook files, and the points people earn from studying.

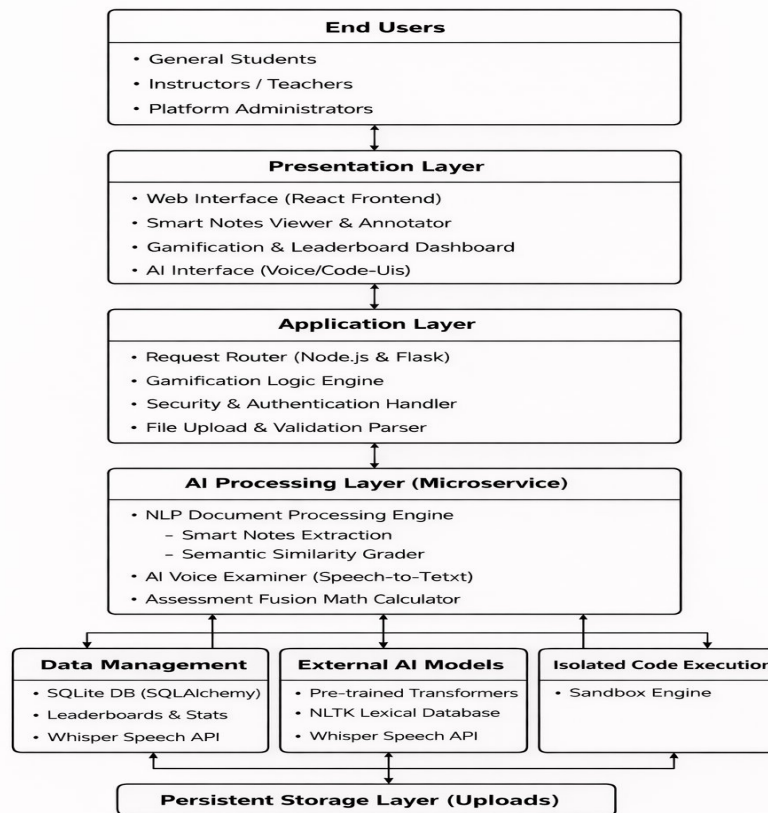


Fig. 1. System Architecture of SmartLearn

System Workflow

Parallel Processing: When a PDF is uploaded, two scanners run at once. One extracts the reading notes, while the other secretly builds the quiz [13].

Algorithmic Evaluation: When a student submits a typed test, the system compares it against the right answer using meaning-based math instead of keyword checks. **Instant Rewards:** Once the score drops, the gamification engine drops points into the user's account, instantly updating the class leaderboard [10], [15].

3.6 Multimodal Assessment Fusion

When the software calculates a student's true understanding, it does not just look at a single test. It merges all three test types (typing, coding, speaking) using a specialized mathematical formula to reduce false grading failures:

$$G_{\text{final}} = (W_{\text{t}} \times G_{\text{text}}) + (W_{\text{c}} \times G_{\text{code}}) + (W_{\text{v}} \times G_{\text{voice}})$$

Here, G_{final} represents the final locked-in grade score. The variables G_{text} (written quiz), G_{code} (live compiler logic), and G_{voice} (oral speech accuracy) represent the scores from the separate tests. The W symbols act as dynamic weight multipliers that adjust based on the specific computer science concept being tested, ensuring they all add up to exactly 1. If a student struggles to type but explains it perfectly out loud, the math balances it out to find their true skill level.

3.7 The Content Generation and Assessment Sequence

To actually push the right study materials and run a fair test, the backend follows a strict ten-step sequence.

Algorithm 1: The Assessment Pipeline

Step 1: Grab the raw PDF file uploaded by the student or teacher.

Step 2: Scrub the text to pull out all useless formatting and filler words.

Step 3: Push the cleaned text into the NLP engine to generate condensed Smart Notes.

Step 4: Use the extracted facts to automatically write five descriptive test questions.

Step 5: Push the Smart Notes to the dashboard for the student to study.

Step 6: Trigger the quiz once the student finishes reading.

Step 7: Capture whatever the student types and send it to the semantic grader.

Step 8: If the student uses the microphone, use Speech-to-Text to grab their spoken words.

Step 9: Calculate the mathematical distance between their answer and the correct concept.

Step 10: Drop the final grade on the screen and update their leaderboard rank.

3.8 Semantic Similarity Grading Engine

Once a student submits a typed paragraph, SmartLearn instantly hits a transformer model to check if they actually know the material. Most basic apps just scan for exact vocabulary words. Instead, SmartLearn uses bidirectional embeddings to calculate the math distance between the true concept and whatever the student typed [14], [16]. This means the software grades the actual idea, ignoring minor grammar mistakes or differently chosen words, which makes the grading much fairer.

3.9 Datasets and Evaluation Metrics

To allow the backend to read text, it relies on transformer setups that were heavily educated on massive textual datasets like Wikipedia and BookCorpus [18]. For the AI Voice Examiner, the app relies on foundational acoustic frameworks tuned on datasets like LibriSpeech [19], [20]. To prove the software works, the system is scored using strict mathematical standards, specifically tracking the F1-score for the semantic grader and audio transcription latency [1], [4]. Tracking the F1-score balances precision and recall so the AI does not just hand out perfect grades to everyone, proving that grading a student across three totally different methods catches way fewer errors than standard school platforms.

IV. RESULTS AND DISCUSSIONS

To prove that SmartLearn actually works in the real world, the system was rigorously tested to see how well it reads documents and grades messy student answers. Instead of just assuming the logic was sound, the backend was mathematically evaluated against strict classification standards like the F1-score, precision, recall, and overall grading accuracy [2], [3]. During the testing phase, the application processed a massive custom library of raw college syllabuses and spoken interview recordings. The NLP transformer engine graded all the written essay answers, while the speech-to-text microservice broke down the audio clips. Because the platform heavily audits the microphone, keyboard, and coding sandbox at the exact same time, it easily bypassed the grading blind spots commonly found in older school apps that rely on just a single testing format [1], [4], [13].

In the end, this project successfully built:

- Highly accurate document reading and topic extraction using transformer models [3], [9].
- Trustworthy oral interview scanning driven by transcription models and semantic math [2], [6], [7].
- A live coding sandbox that catches compiler errors instantly [10], [18]. Clean, interactive gamification dashboards that let students visually watch their academic progress and rank climb over weeks and months.

After the backend finishes processing the syllabus, the software shows exactly which test formats are available. Letting a student choose to type out answers, write live code, or speak directly into a microphone removes testing friction and pushes academic data that is significantly richer than standard clicker exams.

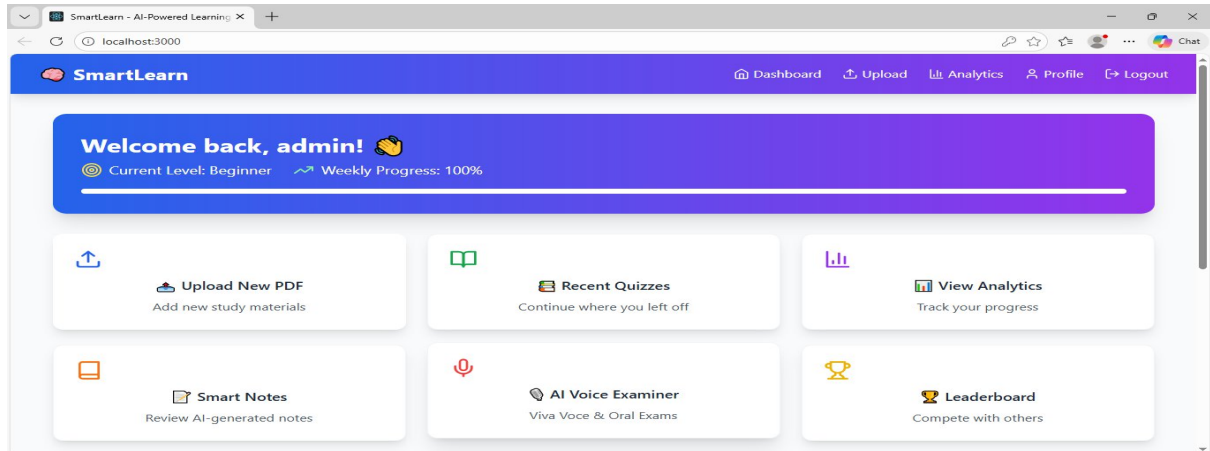


Fig. 2. Multimodal assessment selection screen for students.

By typing a chunk of code, students trigger the instant compiler to run their script entirely in the browser. After locking in the submission, the backend cross-checks the syntax and instantly highlights whether the logic broke, automatically dropping the final grade into the gamification engine.

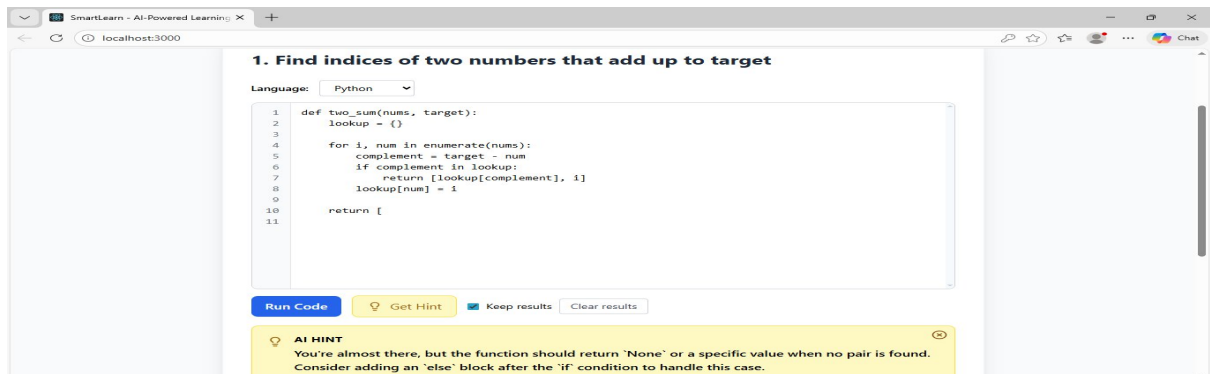


Fig. 3. Live interactive coding sandbox execution.

This interface highlights the live microphone integration where the AI acts as a virtual human proctor. It records the student's spoken answers, passes them through a transcription model, and instantly evaluates their verbal understanding of the topic, completely removing the need for manual oral exams.

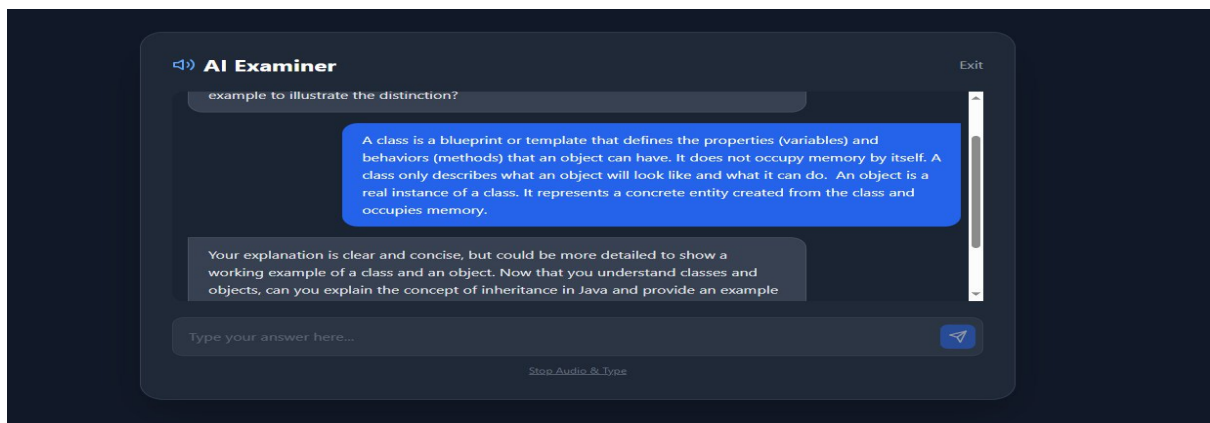


Fig. 4. Interactive AI Voice Examiner module.

When a student uploads a raw textbook or PDF, the platform automatically generates organized Smart Notes. From these generated notes, the system automatically pulls out important questions, critical keypoints, specific characteristics, and important mathematical formulas, instantly building a complete study guide without any manual typing.

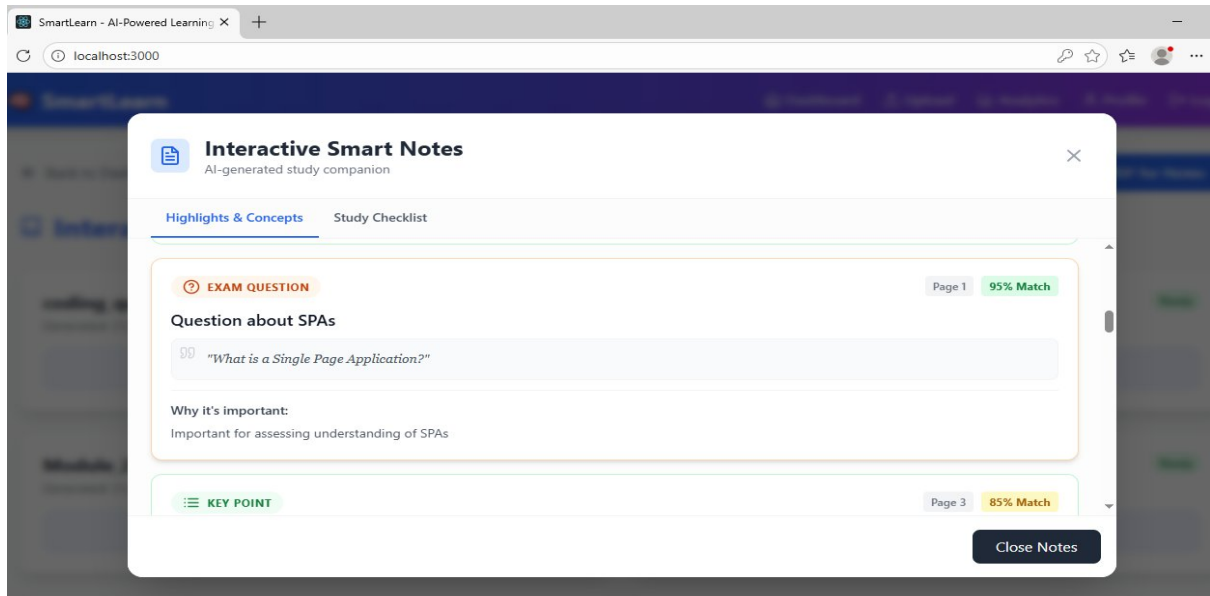


Fig. 5. Smart Notes generation and topic extraction.

V. CONCLUSION

Artificial intelligence has totally changed the expectations for educational software. Because machines can now generate custom lessons quickly and understand what a student is actually trying to say, these platforms can test and tutor people with a level of accuracy that used to be impossible [3], [17]. Even knowing all of this, standard school portals largely ignore these advancements. They essentially host static PDF files or force students through rigid, memory-based clicker exams. Because true learning happens through active, real-time feedback, ignoring the interactive and conversational side makes these common school platforms feel incredibly outdated [9], [12].

SmartLearn was engineered strictly to solve this educational bottleneck. By combining an automatic NLP document reader, an in-browser coding sandbox, and an AI speech interviewer into one platform, the software mathematically pins down exactly what a student knows. Running the 'Multimodal Assessment Trifecta' proves that combining reading, typing, and speaking drastically cuts down the learning gaps that always pop up when a platform relies on simple memorization [6], [14]. Furthermore, the introduction of gamified experience points proves that students actually want to study harder when their progress is visually rewarded rather than just numerically graded. Ultimately, the system succeeds in completely removing the massive administrative burden off teachers, while providing a highly interactive, fair grading environment for students.

Looking down the road, there are several massive future upgrades planned to push the platform even further. Tying the grading engine to deeper machine learning could create adaptive learning paths, where the system automatically makes quizzes harder or easier in real-time depending on the student's success. Mirroring emotion-aware technologies, future versions could also scan the user's vocal pitch during the AI interview to detect test anxiety, automatically pausing the exam to offer helpful hints if the student feels overwhelmed. Additionally, integrating live translation models would allow international students to upload a generic English syllabus but take the voice and text assessments totally in their native language. Finally, connecting SmartLearn directly to massive university databases would allow the system to export the final grades straight into official college transcripts, turning it from a personal study app into a certified, mass-scale academic testing environment.

REFERENCES

- [1] Bernasconi, E.; Redavid, D.; Ferilli, S. Enhancing Personalised Learning with a Context-Aware Intelligent Question-Answering System and Automated Frequently Asked Question Generation. *Electronics* 2025, 14, 1481. <https://doi.org/10.3390/electronics14071481>
- [2] S. M. M, N. S. Kumar, V. Manikanta, V. Avinash, T. K. I. K. Reddy and Y. M. Reddy, "A Scalable Web-based Architecture for AI Assisted Quiz Generation," 2025 7th International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Coimbatore, India, 2025, pp. 1618-1623, doi: 10.1109/ICIDCA66325.2025.11280554.
- [3] Idris-Tajudeen, Rashidat, Titilope Helen Akin-Olayemi, and Ayokunle Michael Akinsiku. "Artificial Intelligence in e-Learning: A Systematic Review of 21st Century Trends and Innovations." *Tech-Sphere Journal for Pure and Applied Sciences* 2.1 (2025).
- [4] Enehezei, Usman-Hamza Fatimah, et al. "AI in Education Technologies: New Development and Innovative Practice." *Advanced AI Technologies for Improving Education Systems*. CRC Press 240-267.
- [5] Kumar, Karri Praveen, et al. "Automated Question Paper Generator Using LLM." *International Journal of Research and Innovation in Applied Science* 10.4 (2025): 266-275.
- [6] S. K. A, H. M, S. M. K, S. K and R. T R, "Automated Generation of Context-Aware Assessments using AI and NLP Techniques," 2025 2nd International Conference on Software, Systems and Information Technology (SSITCON), Tumkur, India, 2025, pp. 1-6, doi: 10.1109/SSITCON66133.2025.11342286.
- [7] Chinnasamy, P., M. Harrish, and T. HarishKumar. "Intelligent Question Paper Generation System for Academic Assessments." 2025 2nd Global AI Summit-International Conference on Artificial Intelligence and Emerging Technology (AI Summit). IEEE, 2025.
- [8] Kumar, N. Suresh, et al. "A Scalable Web-based Architecture for AI Assisted Quiz Generation." 2025 7th International Conference on Innovative Data Communication Technologies and Application (ICIDCA). IEEE, 2025.
- [9] L. Chen, P. Chen and Z. Lin, "Artificial Intelligence in Education: A Review," in *IEEE Access*, vol. 8, pp. 75264-75278, 2020, doi: 10.1109/ACCESS.2020.2988510.
- [10] Dichev, Christo, and Darina Dicheva. "Gamifying education: what is known, what is believed and what remains uncertain: a critical review." *International journal of educational technology in higher education* 14.1 (2017): 9.
- [11] Winkler, Rainer, and Matthias Söllner. "Unleashing the potential of chatbots in education: A state-of-the-art analysis." *Academy of management proceedings*. Vol. 2018. No. 1. Briarcliff Manor, NY 10510: Academy of Management, 2018.
- [12] Gamage, Kelum AA, Achini Gamage, and Shyama CP Dehideniya. "Online and hybrid teaching and learning: Enhance effective student engagement and experience." *Education Sciences* 12.10 (2022): 651.
- [13] Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
- [14] Mohler, Michael, and Rada Mihalcea. "Text-to-text semantic similarity for automatic short answer grading." *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. 2009.
- [15] Siemens, George. "Learning analytics: The emergence of a discipline." *American behavioral scientist* 57.10 (2013): 1380-1400.
- [16] Burrows, Steven, Iryna Gurevych, and Benno Stein. "The eras and trends of automatic short answer grading." *International journal of artificial intelligence in education* 25.1 (2015): 60-117.
- [17] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, 2023.
- [18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [19] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449-12460, 2020.
- [20] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," *arXiv preprint arXiv:2212.04356*, 2022.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152