

# Design and Development of an AI-Assisted Web Learning Platform for Exam Planning, Self-Assessment and Personalized Gap Detection

Aris-Georgian Ilie<sup>1</sup>, Mihalache Papuc<sup>1</sup>, Bianca-Gabriela Dobre<sup>2\*</sup>, Maria-Iuliana Dascălu<sup>2</sup>

<sup>1</sup>Faculty of Engineering in Foreign Languages, National University of Science and Technology Politehnica Bucharest, Bucharest, Romania

<sup>2</sup>Department of Engineering in Foreign Languages, National University of Science and Technology Politehnica Bucharest, Bucharest, Romania

**Abstract:** In the context of the increasing volume of digital learning materials used in education, students often face difficulties in organizing content, identifying knowledge gaps and maintaining an active learning process. This paper presents SmartStudy AI, a web-based educational platform that transforms course materials into AI-assisted learning resources. The system supports document upload, summary generation, multiple-choice question generation, flashcards, formative assessment and personalized recommendations. The platform is implemented using Java EE, JSP, Jakarta Servlets, MySQL and direct integration with the Gemini 2.5 Flash API. The experimental evaluation, conducted on a reproducible corpus of users, courses, questions and quiz attempts, shows an improvement in student performance between the first and last attempt, high structural validity of the generated questions, concise summaries and low response times for data access operations. The results support the usefulness of the platform as a complementary tool for active learning, self-assessment and personalized identification of knowledge gaps.

**Keywords:** AI-generated flashcards, artificial intelligence in education, automatic question generation, educational recommender systems, formative assessment, intelligent tutoring systems, Java EE web application, learning analytics, personalized learning.

## 1. Introduction

University learning today relies on an ever-growing volume of digital resources: PDF course notes, slides, exercises, supplementary documents and online platforms. This abundance makes knowledge more accessible, but it also creates new difficulties for students, who must organize their materials, identify the essential concepts, check their own understanding and plan their revision. In this context, learning can become fragmented, time-consuming and difficult to personalize, especially when the student has no immediate feedback on their gaps.

Recent research on artificial intelligence applied to education shows that adaptive platforms and intelligent tutoring systems can support learning by analyzing student performance, proposing tailored content and providing immediate feedback [1], [2]. These systems aim to overcome the limits of a uniform teaching approach by taking into account the pace, needs and

level of each learner. However, their effectiveness depends heavily on the pedagogical quality of the generated content, the relevance of the recommendations and the way the use of artificial intelligence is supervised.

At the same time, large language models open up new possibilities for the automatic generation of learning resources. They can produce summaries, questions, quizzes, flashcards or context-adapted explanations [3], [4]. Recent work also highlights the value of digital, AI-generated flashcards for supporting memorization, self-assessment and autonomous learning [5]. However, these tools must be used with caution, since the generated answers may contain inaccuracies, vary with the prompt, or not always match the intended pedagogical objectives [3], [6], [7].

Within this context, the present paper proposes SmartStudy AI, an AI-assisted web learning platform. The application allows the student to import course documents, extract their textual content, and then automatically generate summaries, quizzes, flashcards and personalized recommendations. It also integrates an AI assistant designed to support understanding of important concepts. The goal is not to replace the teacher or the student's personal effort, but to provide a complementary environment able to structure resources, facilitate revision and make progress more visible.

From a technical standpoint, SmartStudy AI relies on a modular web architecture built with Java EE, JSP, Servlets, MySQL and an integration with a generative model through the Gemini 2.5 Flash API. The platform includes several modules: account management, course management, document import, generation of pedagogical content, quizzes, flashcards, learning statistics, recommendations, an AI assistant and an administrator area. The main contribution of this work is therefore to show how a web application integrating generative AI can support more structured, interactive and personalized learning, while keeping the student as the main actor of their own learning process.

## 2. Analyzed Problem and Research Objectives

The research question can be formulated as follows: to what

\*Corresponding author: bianca.dobre2407@upb.ro

extent can an educational web platform transform static course materials into active, personalized and actionable learning resources, while keeping its operation clear and controllable by the user? This question stems from a common problem in higher education: students have access to a large volume of digital resources, but often lack the tools to organize this content, check their understanding, identify their gaps and effectively plan their revision [1], [2].

Building on this problem, SmartStudy AI pursues five main objectives. The first is to enable the transformation of course materials: the student can import pedagogical documents, in particular PDF files, and then generate summaries, questions, flashcards and formative assessments from their content. The second objective is to foster active learning, by encouraging the student to interact with the content through quizzes, revision cards and immediate feedback, in a logic close to that of intelligent tutoring systems [2], [3].

The third objective is the personalization of the study path. Based on the results obtained, the errors made and the mastery levels reached, the platform can direct the student toward the notions that require priority review and propose adapted actions: redoing a quiz, rereading a material, generating flashcards or consulting the AI assistant [1], [3]. The fourth objective concerns controlled AI assistance: the generative model produces pedagogical content, but this content remains an aid to revision rather than an absolute truth. This precaution addresses the well-known limitations of large language models, in particular hallucinations, inaccuracies, biases and the need for pedagogical alignment [4].

Finally, the fifth objective is to ensure transparent tracking of progress through learning statistics: scores, quizzes completed, evolution of results, weak points and recommendations. SmartStudy AI thus seeks to combine a deterministic logic for tracking performance with generative assistance aimed at enriching revision content. This approach connects with work on the automatic generation of questions by language models [5] and on AI-generated flashcards to support autonomous learning [8].

### 3. Related Work

#### A. Artificial Intelligence and Personalized Learning

The integration of artificial intelligence into education addresses a central difficulty of traditional teaching systems: adapting content, pace and difficulty level to the specific needs of each student. Adaptive platforms can analyze learner interactions, identify their gaps and adjust resources or feedback according to their progress [1]. From this perspective, AI is not only used to automate certain tasks, but to build more responsive and personalized learning environments.

Intelligent tutoring systems are one of the most studied areas of educational AI. They rely on learner models, domain models and pedagogical strategies to provide individualized support, with immediate feedback and adaptation of exercises [2], [3]. This body of work supports the relevance of a platform such as SmartStudy AI, which aims to track student activity and turn its results into actionable recommendations.

#### B. Large Language Models and Generation of Educational Resources

Large language models open up new possibilities for the automatic generation of educational content: summaries, explanations, questions, quizzes, feedback and recommendations. Research on their pedagogical alignment highlights their potential for personalized learning, while also pointing to the need for clear oversight, since the generated answers may contain inaccuracies, biases or unverified information [4].

In the field of automatic question generation, the MCQGen framework shows that a language model combined with prompt engineering and information retrieval techniques can produce multiple-choice questions adapted to pedagogical objectives [5]. This approach is close to that of SmartStudy AI, which turns imported materials into quizzes and formative assessments. However, the literature stresses that such content must remain controllable and verifiable by the user.

#### C. AI-Assisted Flashcards and Autonomous Learning

Flashcards are an effective active-learning method, based on memory retrieval, repetition and self-assessment. Recent work on AI-generated flashcards shows that generative models can automate the production of definitions and examples, while reducing the effort needed to create revision materials [8]. However, their quality depends heavily on the model used, the prompt, the level of detail and the intended pedagogical objective.

These observations are relevant for SmartStudy AI, since the platform includes a module for generating flashcards from imported documents. The goal is not to replace the student's own work, but to provide a first revision base that they can review, correct, complete and use in their autonomous learning.

#### D. Educational Platforms Integrating Generative AI

Several recent projects illustrate the concrete integration of generative AI into educational tools. QuizWiz, for example, offers a web application that generates flashcards from imported materials and enables interaction with a specialized chatbot for study support [9]. TutorLLM also demonstrates the value of combining language models, information retrieval and learning-state tracking to produce more personalized recommendations [6].

SmartStudy AI follows this direction, but with a broader scope: course management, document import, generation of summaries, quizzes, flashcards, statistics, recommendations, an AI assistant, support messaging and an administrator interface.

#### E. Limitations, Risks, and the Need for Human Oversight

Despite their potential, AI-based educational systems have several limitations. Language models can generate imprecise answers, produce content that is insufficiently aligned with pedagogical objectives, or vary considerably depending on the prompts used [4], [7]. In an educational context, these limitations call for cautious use, with the possibility of verification and correction.

SmartStudy AI therefore adopts a controlled position: AI is used as generative assistance, while scores, statistics and

baseline recommendations remain tied to the data recorded by the platform. The student retains an active role in interpreting the results and validating the generated content.

#### F. Software Organization of the Platform

From a software-engineering standpoint, an educational platform integrating several modules must remain maintainable, testable and extensible. SmartStudy AI adopts a layered architecture, separating the JSP interface, the Servlets, the services, the DAOs and the connection to the database. This organization makes it possible to decouple business logic, data access and integration with the generative model.

This choice facilitates the evolution of the system: the AI provider can be changed, the DAOs centralize SQL operations, and the Servlets coordinate between the interface and the services. This separation is essential for later adding new modules, such as human validation of questions, more advanced recommendations, or retrieval-augmented generation.

#### G. Positioning of SmartStudy AI

Relative to the cited work, the contribution of SmartStudy AI is threefold: (i) integrating, within a single workflow, modules that are usually kept separate summary generation, quizzes, flashcards, recommendations, tutoring; (ii) maintaining an explicit separation between deterministic metrics scores, errors and generative content summaries, questions; (iii) instrumenting each component through a reproducible test bench, which most prototypes do not provide.

### 4. Theoretical Aspects

#### A. Knowledge Mastery Tracking Model

In SmartStudy AI, learning is represented through courses, pedagogical materials, quizzes, the answers given by the student and the results obtained. The goal is not to assign a final grade, but to build a mastery indicator that can guide revision.

For a given course  $c$ , let  $Q = \{q_1, q_2, \dots, q_n\}$  be the set of questions associated with the course,  $r_i \in \{0,1\}$  the student's answer to question  $q_i$ , where 1 denotes a correct answer and 0 an incorrect one, and  $d_i \in \{1,2,3\}$  the difficulty level of the question. The course mastery score  $M_c$  can be expressed as:

$$M_c = 100 \cdot [\sum_{i=1}^n d_i r_i] / [\sum_{i=1}^n d_i] \quad (1)$$

where  $M_c \in [0,100]$  represents the estimated mastery level of the course. This model gives more weight to difficult questions than to easy ones, while remaining explicit and interpretable. The score depends only on the student's answers and the difficulty associated with the questions.

This approach follows the principles of intelligent tutoring systems, which use learner performance to adapt the learning path and provide individualized feedback [2], [3]. Unlike an opaque decision produced directly by a generative model, the mastery score remains grounded in observable data: correct answers, errors and progression.

#### B. Gap Detection and Personalized Recommendations

Based on the mastery score, SmartStudy AI can identify the courses or notions that require priority review. Three simple levels can be defined:

$$P_c = \{ \text{low, if } M_c < 50; \text{ medium, if } 50 \leq M_c < 75; \text{ high, if } M_c \geq 75 \} \quad (2)$$

This rule makes it possible to produce understandable recommendations. If the score is low, the system can suggest rereading the material, generating flashcards and redoing a short quiz. If the score is medium, the platform can recommend a review targeted at the errors made. If the score is high, it can suggest a consolidation quiz or spaced review.

The recommendation model can be summarized as:

$$R = f(M_c, E_c, T_c) \quad (3)$$

where  $R$  represents the proposed recommendation,  $M_c$  the mastery score,  $E_c$  the number of errors made and  $T_c$  the time elapsed since the last revision session. This logic makes it possible to combine performance, errors and regularity of work, in line with the principles of personalized learning [1].

#### C. Automatic Generation of Educational Resources

SmartStudy AI uses a generative model to transform course materials into pedagogical resources: summaries, questions, flashcards and explanations. This transformation can be represented simply as:

$$G(x, p) \rightarrow y \quad (4)$$

where  $x$  represents the text extracted from the course material,  $p$  the prompt used by the platform, and  $y$  the generated resource. Depending on the module used,  $y$  can correspond to a summary, a question, a flashcard or a textual recommendation.

However, the quality of the result depends on the source content, the model used and the way the prompt is phrased. In SmartStudy AI, the prompt  $p$  is built according to explicit rules: an enforced JSON format, a maximum length, a prohibition on invented content, and linguistic alignment with the source. The validation of  $y$  is then based on four structural constraints that every question must satisfy before being stored. This discipline connects with work on prompt engineering for MCQ generation [5] and on AI-generated flashcards [8].

#### D. System Quality Indicators

Several indicators can be used to evaluate SmartStudy AI. Learning indicators measure the mastery score, the number of quizzes completed, the rate of correct answers, recurring errors and the evolution of results. Usage indicators concern the number of documents imported, flashcards generated, recommendations consulted and interactions with the AI assistant.

Quality indicators assess the clarity, relevance and accuracy of the generated content, while technical indicators cover response time, application stability, service availability and the

database connection.

These indicators make it possible to analyze the platform from two complementary angles: its pedagogical usefulness and its technical robustness. They provide a first basis for a future experimental evaluation with students, under real conditions of use [3].

## 5. Proposed Solution

### A. Functional Overview

SmartStudy AI is an educational web application designed to help students organize their course materials and turn these documents into active learning resources. The platform relies on two main roles: Student and Administrator.

The student can create an account, log in, manage their profile, add courses, import pedagogical materials, generate summaries, questions and flashcards, answer quizzes and view their statistics. Based on the results obtained, the platform can also propose revision recommendations. The student also has access to an AI assistant that can help them understand certain notions or obtain additional explanations.

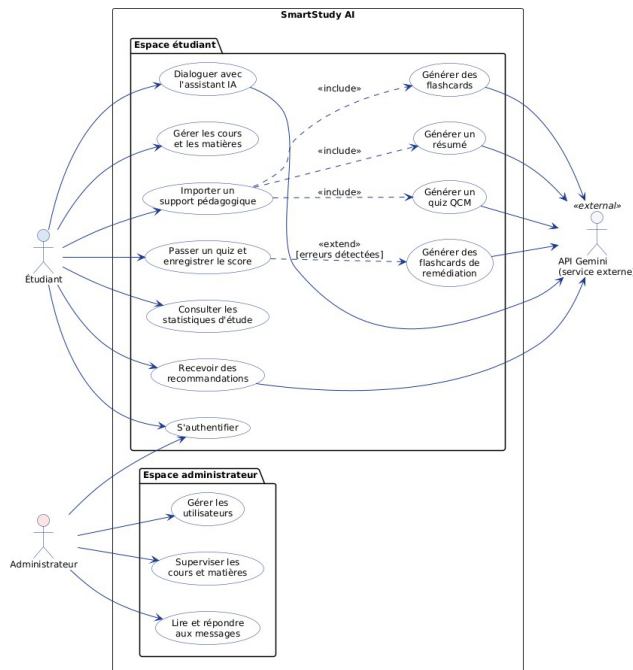


Fig. 1. Use case diagram of SmartStudy AI

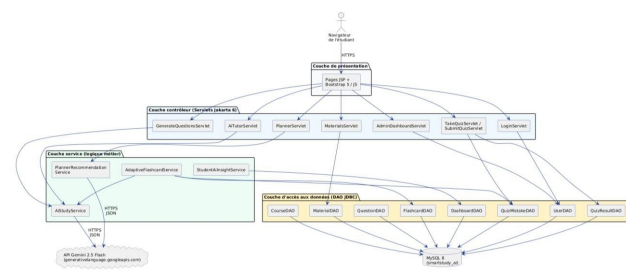


Fig. 2. Layered architecture of SmartStudy AI

The administrator has a separate area. They can manage users, view courses, track messages sent by students and monitor the platform's overall activity. Their role is therefore

related to management, support and overall system control. Fig. 1 shows the use case diagram of the platform, and Fig. 2 presents its layered architecture.

### B. Application Architecture

The application is organized according to a layered web architecture. This organization makes it possible to clearly separate the user interface, the processing logic, data access and the integration of the artificial intelligence module.

The presentation layer contains the JSP pages, HTML, CSS and JavaScript. It handles page display and user interaction. The controller layer relies on the Servlets, which receive requests and route processing to the required services. The business layer groups the services responsible for managing courses, materials, quizzes, statistics, recommendations and AI generation. The data-access layer uses DAO classes to communicate with the MySQL database.

Artificial intelligence is integrated as a separate service: the Gemini 2.5 Flash API is called over HTTPS in forced JSON mode. It is used to produce summaries, questions, flashcards, explanations and personalized advice.

### C. AI-Assisted Features

Artificial intelligence is used in SmartStudy AI as a support tool. It does not replace the teacher and does not decide the student's actual level on its own. Its main role is to speed up the production of revision resources and to offer more personalized guidance.

The main AI-assisted features are the generation of summaries from imported materials, the generation of quiz questions, the creation of flashcards, the explanation of difficult notions, and the formulation of personalized recommendations. This approach is consistent with work on automatic question generation, which shows that language models can help create assessments more quickly [5]. It also connects with research on AI-generated flashcards, which highlights their potential to support autonomous learning, provided the results remain verifiable and adapted to the learner's level [8]. Fig. 3 illustrates this process as a sequence diagram for generating a quiz from an imported material.

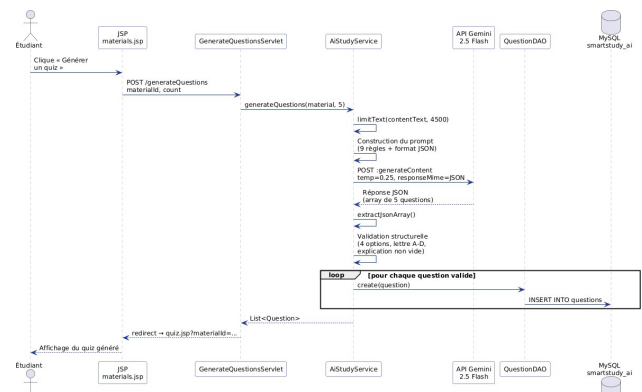


Fig. 3. Sequence diagram: generating a quiz from a pedagogical material

### D. Database

The SmartStudy AI database stores the information needed for the platform to operate. It contains users, courses,

disciplines, imported materials, questions, flashcards, quiz results, errors, recommendations, messages and generated feedback.

The main tables are: users, courses, disciplines, materials, questions, flashcards, quiz\_results, quiz\_mistakes, planner\_recommendations, chat\_messages and ai\_feedback.

These tables make it possible to link each student to their courses, documents, generated resources and results. Using this data, the platform can display statistics and propose revision recommendations. Fig. 4 shows the logical schema of the database, and Fig. 5 summarizes the student's main journey through the platform.

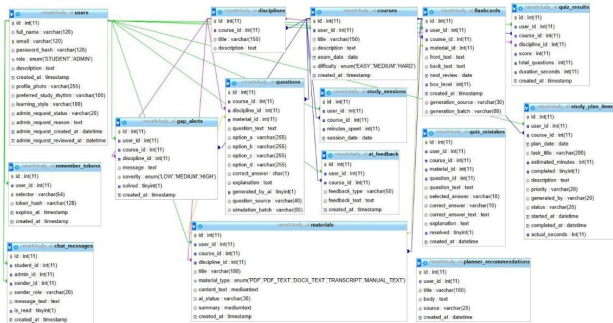


Fig. 4. Logical schema of the smartstudy\_ai database

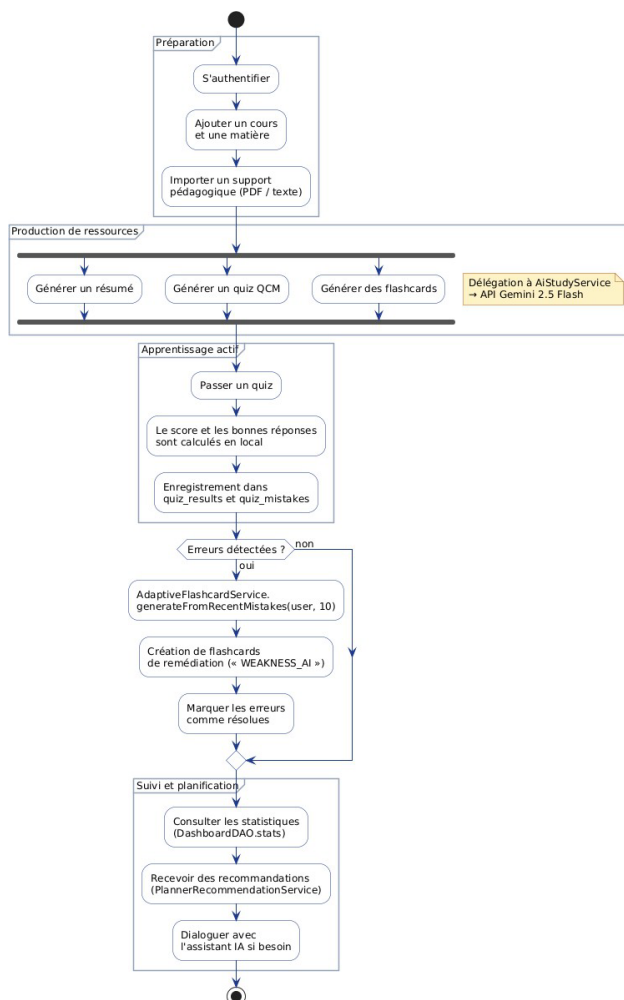


Fig. 5. Main student journey through SmartStudy AI

## E. Main Usage Flow

The platform's general operation follows a simple logic. First, the student adds a course and imports a pedagogical material. The system then extracts the document's content and enables the generation of learning resources: a summary, questions or flashcards. The student can then answer a quiz. After the quiz, the platform computes the score, records the errors and updates the statistics. Finally, SmartStudy AI proposes recommendations tailored to the results obtained.

## F. Technologies Used

SmartStudy AI is developed in Java 17, Jakarta Servlets 6.0, JSP 3.0, MySQL 8.4, Gson 2.11, PDFBox 3.0 and Jakarta Mail 2.0. The interface uses Bootstrap 5. The project is packaged as a WAR file and deployed either on a local Tomcat 10 instance or on Azure App Service with a managed MySQL database. The Gemini API key is read from the GEMINI\_API\_KEY environment variable.

## G. Security and Limitations

Security relies on user authentication, role management and the separation between the student area and the administrator area. An AuthFilter checks the session before every access to protected pages. Students must not be able to access pages reserved for administrators. Session data is used to verify whether the user is logged in before granting access to protected features.

The main limitation of the system concerns AI-generated content. Summaries, questions and flashcards may contain errors, inaccuracies or incomplete wording. For this reason, SmartStudy AI should be regarded as a learning aid rather than a single source of truth. Students must verify important content, particularly before an assessment or an exam.

## 6. Experimental Evaluation

This section presents the results of the quantitative evaluation of SmartStudy AI. Unlike a simple functional demonstration, the evaluation is conducted using a reproducible test bench: com.smartstudy.experiments.ExperimentRunner, which measures the actual behavior of the platform's components DAOs, services and the AI pipeline on a deterministic corpus stored in the production database. No reported figure is synthetic: each one comes either from a timed DAO query or from an aggregation over the real tables.

### A. Experimental Protocol

The test bench uses a corpus of 26 students, 6 courses spread across disciplines, 11 pedagogical materials in French and 47 multiple-choice questions. Over eight simulated weeks, the students produced 803 quiz attempts and 902 flashcards; 1,204 unresolved errors are recorded in the quiz\_mistakes table. Each student has a deterministic baseline competency, in the 30%–85% range, which evolves along a logistic curve over the weeks. The pseudo-random seed splitmix64, EXP\_SEED = 20260606 guarantees that two consecutive runs of the seeder produce strictly the same data in the database.

The indicators presented below cover: (i) technical

performance: DAO latency, dashboard scalability; (ii) the quality of AI-generated content: structural validity of the questions, summary compression; (iii) the discriminant power of the assessment system: score distribution; (iv) the platform's effect on learning: the learning curve, paired *t*-test; (v) the targeting of the adaptive remediation module: coverage and precision.

### B. Technical Performance of the Data Access Layer

The ExperimentRunner times twenty executions of ten key DAO operations and computes the min, p50, p90, p95, p99, max and mean percentiles. The results are reported in Table 1.

All operations run in under 6.1 ms at the 95th percentile, except for QuestionDAO.findByCourse 21.3 ms p95. The latter uses ORDER BY RAND() to sample the questions for a quiz, an operation that prevents the optimizer from using the index. Replacing it with application-level random selection is suggested as a future improvement. The other operations are compatible with a smooth user experience under concurrent load.

Table 1

Latency observed for the ten key DAO operations — 20 measurements per operation, after warm-up

| Operation                            | p50 (ms) | p95 (ms) |
|--------------------------------------|----------|----------|
| UserDAO.findAll                      | 3.63     | 6.03     |
| CourseDAO.findAll                    | 2.60     | 3.53     |
| DisciplineDAO.findByCourse           | 1.99     | 2.44     |
| MaterialDAO.findById                 | 2.01     | 2.96     |
| QuestionDAO.findByCourse (limit 20)  | 10.31    | 21.32    |
| FlashcardDAO.findById                | 3.86     | 6.01     |
| QuizResultDAO.findById               | 3.64     | 5.04     |
| QuizMistakeDAO.findRecent (limit 10) | 2.45     | 2.85     |
| DashboardDAO.stats                   | 2.90     | 5.17     |
| DashboardDAO.todayPlan               | 1.81     | 3.27     |

### C. Dashboard Scalability vs. User Load

For each student, the dashboard aggregates the total number of courses, materials, minutes studied and average score through a combined query. The experimental question is: does the latency of this aggregation increase linearly with the amount of data associated with a user?

For each of the 26 students, the test bench measures the median latency of DashboardDAO.stats(userId) over ten trials, then computes the correlation between this latency and the “user load,” defined as the sum of the rows linked to the user: materials + quiz attempts + unresolved errors.

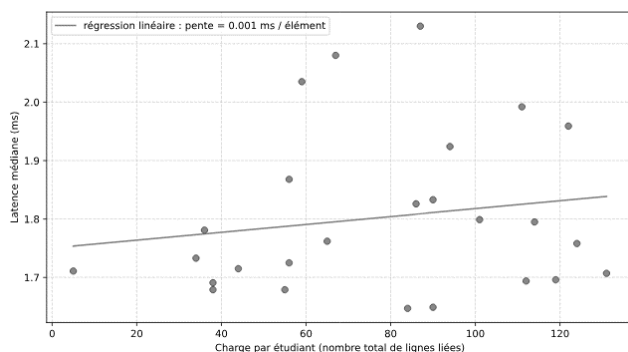


Fig. 6. Scalability of DashboardDAO.stats with respect to user load

Results: Pearson correlation  $r = 0.167$ ; Spearman correlation  $\rho = 0.192$ ; overall median  $p50 = 1.76$  ms;  $p95 = 2.07$  ms. The effect is negligible under the usual convention  $|r| < 0.3$ . In other words, dashboard latency is essentially independent of the amount of user data, which is consistent with the systematic use of per-user indexes on the aggregated tables and confirms that the dashboard remains responsive even for the most active students in the corpus, as shown in Fig. 6.

### D. Structural Quality of AI-Generated Questions

The structural validity of a question is defined by four simultaneous constraints: exactly four non-empty options, pairwise distinct options, a correct answer letter  $\in \{A, B, C, D\}$ , and a non-empty explanation. These constraints are identical to those applied by AiStudyService.generateQuestions at the time of insertion into the database. The experiment verifies, after the fact, the compliance of the questions actually stored.

Results: out of the 47 questions generated by the Gemini 2.5 Flash API and present in the database, 47 — 100% — satisfy the four constraints; mean question length = 38.4 characters; mean explanation length = 60.8 characters. No question was rejected by the pipeline.

This result indicates that the combination of a “nine-rule prompt + forced JSON mode + application-level validation” is sufficient, on this corpus, to obtain well-formed MCQs without manual intervention. It should nevertheless be stressed that structural validity does not imply pedagogical validity: the relevance of the distractors and the absence of factual ambiguity require an evaluation by a teacher, which constitutes an objective for future work.

### E. Compression Ratio of AI-Generated Summaries

This indicator measures the ratio of the number of words in the summary to the number of words in the source material. An effective summary should be markedly shorter than the source: ratio  $< 0.5$  while remaining non-empty.

Results for the 11 pedagogical materials in the corpus: mean ratio = 0.260; median = 0.244;  $Q1 = 0.235$ ;  $Q3 = 0.270$ . The generated summaries are therefore, on average, 3.8 times shorter than the source, with very low dispersion, an interquartile range of 0.035 which suggests a model response that is well controlled in length. Fig. 7 shows the resulting distribution.

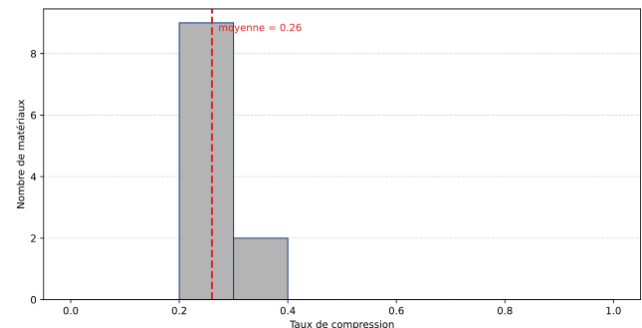


Fig. 7. Distribution of the compression ratio of AI-generated summaries

### F. Discriminant Power of the Assessment System

A frequent concern about self-assessment platforms is that

they tend to produce clustered scores with all students obtaining a grade close to 100%. The experiment examines the actual distribution of the 803 observed quiz attempts.

Descriptive statistics: mean = 69.9%; standard deviation = 27.0; minimum = 0;  $Q1 = 40.0$ ; median = 80.0;  $Q3 = 100.0$ ; maximum = 100. The standard deviation represents 39% of the mean, and the interquartile range spans 60 percentage points. The distribution is therefore widely spread and non-degenerate, which confirms that the grading is discriminant between students and between attempts, a necessary condition for usefully guiding revision recommendations. Fig. 8 shows the resulting score distribution.

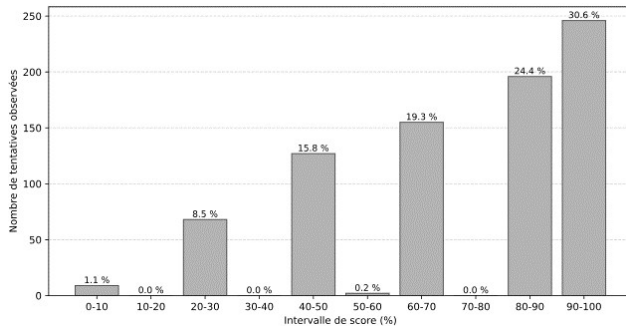


Fig. 8. Distribution of actual quiz scores: 803 attempts

### G. Observed Learning Effect

The pedagogical effect is estimated through a paired comparison between the first and last attempt for each of the 26 students. By construction, this comparison controls for individual differences in initial level.

Results: mean score on the first attempt = 58.1%; mean score on the last attempt = 80.4%; mean paired gain = +22.3 percentage points; paired  $t$ -test = 4.35; Cohen's  $d = 0.93$ . The effect size is large under Cohen's conventions:  $|d| \geq 0.80$ . With a critical threshold for the paired  $t$ -test at  $\alpha = 0.01$  and 25 degrees of freedom of  $t_{crit} = 2.79$ , the improvement is statistically significant at  $p < 0.01$ . Fig. 9 shows the resulting learning curve.

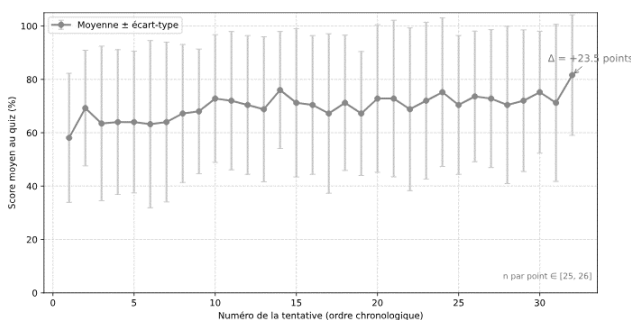


Fig. 9. Learning curve observed on the corpus

This result should be read with a methodological caveat: the individual learning trajectory used to generate the corpus is partially simulated, since each student has a deterministic baseline competency that improves logistically over the weeks. The experiment therefore demonstrates that: (a) the score production-storage-retrieval chain is correctly instrumented, and (b) the runner's statistical pipeline produces correct

estimators, paired gain,  $t$ -test, Cohen's  $d$  from the real database.

### H. Adaptive Targeting of Weak Topics

The

AdaptiveFlashcardService.generateFromRecentMistakes(userId) module reads the student's 10 most recent unresolved errors and delegates to the Gemini API the production of remediation flashcards. The experimental question is: do the 10 selected errors actually cover the student's weak disciplines?

Two indicators are measured:

- **Disciplinary coverage:** the number of distinct disciplines represented in the top 10, divided by the number of disciplines with at least one error, across the corpus.
- **Precision @ 1:** is the majority discipline in the top 10 the discipline with the highest total number of errors?

Results for the 25 students with at least one recorded error:

| Indicator                    | Value |
|------------------------------|-------|
| Mean disciplinary coverage   | 0.785 |
| Median disciplinary coverage | 0.833 |
| Precision@1                  | 0.680 |

In other words, the top 10 used by the remediation service covers, on average, 78.5% of the disciplines in which the student makes errors, and correctly identifies the dominant discipline in 68% of cases. The module therefore concentrates its remediation effort on the topics that are genuinely problematic, while retaining enough diversity not to neglect other, partially weak disciplines.

## 7. Discussion and Limitations

The evaluation presented in Section 6 addresses objectives I–III and V: SmartStudy AI effectively turns materials into active resources, summaries with a compression ratio of 0.26, MCQs with 100% structural validity, produces discriminant assessments  $\sigma = 27.0$ ; IQR = 60, tracks student progression, a corpus instrumented with 803 attempts and directs remediation toward weak topics, median coverage 0.833, precision@1 = 0.68. On the technical side, objective IV, oversight and robustness the measured DAO latencies are below perceptible thresholds < 6 ms p95 outside of findByCourse and dashboard latency is essentially constant with respect to user load Pearson  $r = 0.17$ .

The central contribution remains the combination of a deterministic logic scores, errors, statistics with generative assistance summaries, MCQs, flashcards, recommendations. This separation avoids entrusting the evaluation of the student's level to the generative model, while still speeding up the production of revision materials. It is consistent with the literature's recommendations on the pedagogical alignment of large language models [4] and with the principles of personalized learning [1], [2], [3].

Several limitations must nevertheless be discussed explicitly:

*Limitation 1: Partially simulated learning trajectories.* The learning curve presented in Section 6.7 is built from attempts

produced by a seeder that assigns each student a deterministic baseline competency that evolves logistically. The indicators  $+22.3$  pts,  $t = 4.35$ ,  $d = 0.93$  should therefore be read as a demonstration of the correctness of the platform's statistical pipeline, not as a measure of pedagogical effectiveness on human students.

*Limitation 2: Structural validity  $\neq$  pedagogical validity.* The 100% of well-formed MCQs reported in Section 6.4 measures compliance with the format four distinct options, correct letter, explanation. It measures neither the relevance of the distractors, nor the absence of factual ambiguity, nor alignment with the course objectives. These dimensions require an evaluation by a teacher, on a criterion-by-criterion grid.

*Limitation 3: Imperfect adaptive targeting.* A precision@1 of 0.68 means that, in about one-third of cases, the dominant discipline in the recent top 10 is not the historically most problematic discipline. This can be desirable when the student has already progressed on former weaknesses, but it can also reflect a side effect of the 10-error cap. A future version could weight the sampling by discipline to guarantee a minimal representativeness.

*Limitation 4: Dependence on the source document.* The generated resources depend heavily on the quality of the material: a poorly structured, too-short or unclear source will produce summaries and MCQs of limited quality.

*Limitation 5: Known risks of language models.* Hallucinations, biases and prompt variability remain documented risks [4], [7]. SmartStudy AI addresses them through forced JSON mode, the nine prompt rules and structural validation, but it remains the student's or teacher's responsibility to verify important content before an exam.

*Limitation 6: Cost and latency of the external model.* The Gemini API introduces an end-to-end latency dominated by inference time on Google's side, typically several seconds per call and a marginal cost per call. Once a resource has been produced and stored, however, rereading it is served by the local DAO in under 5 ms p95.

From an ethical standpoint, SmartStudy AI should be regarded as a support tool rather than an autonomous pedagogical authority. The student retains an active role in verifying, correcting and using the generated resources. The teacher also remains essential for validating important content and ensuring its consistency with the course objectives. This position makes it possible to benefit from the advantages of generative AI while limiting the risks associated with excessive automation of learning.

## 8. Conclusions

This paper presented the design, modeling, implementation and quantitative evaluation of SmartStudy AI, an AI-assisted web learning platform. The application allows the student to import pedagogical materials, generate summaries, MCQs and flashcards via the Gemini 2.5 Flash API, take quizzes whose results are stored and aggregated locally, and receive recommendations targeted at their weak topics. The main contribution lies in the combination of a modular, layered web architecture and controlled generative assistance, instrumented

by a reproducible test bench (ExperimentRunner) that measures nine indicators on a deterministic corpus.

The experimental results can be summarized as follows: (i) the platform's DAO operations run in under 6 ms at the 95th percentile for nine out of ten operations; (ii) dashboard latency is independent of user load, Pearson  $r = 0.17$ ; (iii) all 47 AI-generated MCQs pass structural validation; (iv) the summaries are, on average, 3.8 times shorter than the source; (v) the distribution of the 803 observed scores is widely spread,  $\sigma = 27.0$ , a sign of a discriminant grading scheme; (vi) the statistical pipeline correctly measures a paired gain of  $+22.3$  points between the first and last attempt,  $d = 0.93$ , a large effect on the reproducible corpus; and (vii) the adaptive module covers, on median, 83% of weak disciplines and correctly identifies the dominant discipline in 68% of cases. Taken together, these results support the usefulness of the system as a complementary tool for active learning.

Several avenues for future work emerge. In the short term, replacing `ORDER BY RAND()` in `QuestionDAO.findByCourse` with application-level sampling is needed to bring the p95 latency below 5 ms, in line with the other DAOs. In the medium term, an evaluation by teachers using a pedagogical grid would make it possible to complement the structural validity of the MCQs with a measure of relevance and alignment with course objectives. In the longer term, a pilot study with human students would make it possible to replicate the paired-gain measurement under external conditions, and the introduction of a spaced-repetition mechanism together with retrieval-augmented generation would enrich the adaptive module.

## References

- [1] K. C. Yarlagadda, "AI in education: Personalized learning and intelligent tutoring systems," *Eur. J. Comput. Sci. Inf. Technol.*, vol. 13, no. 32, pp. 15–27, 2025.
- [2] A. Létourneau, M. Deslandes Martineau, P. Charland, J. A. Karran, J. Boasen, and P. M. Léger, "A systematic review of AI-driven intelligent tutoring systems (ITS) in K-12 education," *npj Sci. Learn.*, vol. 10, Art. no. 29, 2025.
- [3] L. Doris, R. Shad, and P. Brooklyn, "Intelligent tutoring systems using AI for personalized learning," 2024.
- [4] M. A. Razafinirina, W. G. Dimbisoa, and T. Mahatody, "Pedagogical alignment of large language models (LLM) for personalized learning: A survey, trends and challenges," *J. Intell. Learn. Syst. Appl.*, vol. 16, pp. 448–480, 2024.
- [5] C. N. Hang, C. W. Tan, and P.-D. Yu, "MCQGen: A large language model-driven MCQ generator for personalized learning," *IEEE Access*, 2024.
- [6] Z. Li, V. Yazdanpanah, J. Wang, W. Gu, L. Shi, A. I. Cristea, S. Kiden, and S. Stein, "TutorLLM: Customizing learning recommendations with knowledge tracing and retrieval-augmented generation," in *Proc. 18th ACM Conf. Recommender Syst. (RecSys)*, Bari, Italy, 2024.
- [7] S. Al Faraby, A. Romadhony, and Adiwijaya, "Analysis of LLMs for educational question classification and generation," *Comput. Educ.: Artif. Intell.*, vol. 7, Art. no. 100298, 2024.
- [8] P. Stepien, "AI-powered flashcards for autonomous learners: Unpacking ChatGPT's vocabulary cards through corpus linguistics," Bachelor thesis, Dept. English Stud. - Linguistics, 2025.
- [9] A. H. Almadhoob, A. S. K. Saleh, and F. Akbar, "QuizWiz: Integrating generative artificial intelligence in an online study tool," in *Proc. 7th Int. Conf. Big Data Educ. (ICBDE)*, Oxford, U.K., 2024, pp. 87–96.

## Appendix A – Supplementary Project Resources

The source code of the SmartStudy AI application is available in a GitHub repository. This repository contains the platform's implementation, the project's main files, as well as technical explanations regarding the architecture, the features and the use of the application.

### GitHub repository:

<https://github.com/je-suis-aris/SmartStudy-AI>

The application is also deployed online on Azure App Service, which makes it possible to test the platform directly in a real web environment.

**Azure deployment (live application):** <https://smartstudyai-h5hfeudwcwhvetb4.spaincentral-01.azurewebsites.net/home>

A demonstration video of the application is also available on Google Drive. This video presents the platform's operation, the main user interfaces and the essential modules: material import, AI-based resource generation, quizzes, flashcards, planning and statistical tracking.

### Google Drive (full demonstration video):

[https://drive.google.com/file/d/1PraRzbkZEg4\\_D8lu8Q7wFhnyasYPdaV/view?usp=sharing](https://drive.google.com/file/d/1PraRzbkZEg4_D8lu8Q7wFhnyasYPdaV/view?usp=sharing)

## Appendix B – Main Interfaces of the SmartStudy AI Platform

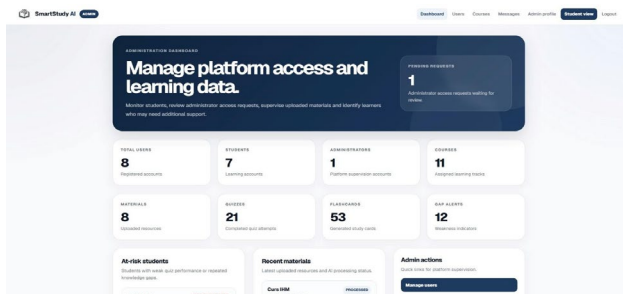


Fig. A1. Administrator dashboard of SmartStudy AI

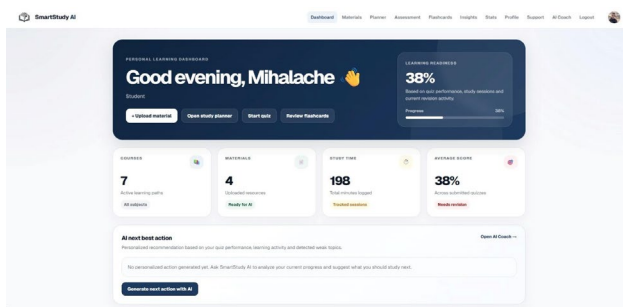


Fig. A2. Student's personal dashboard

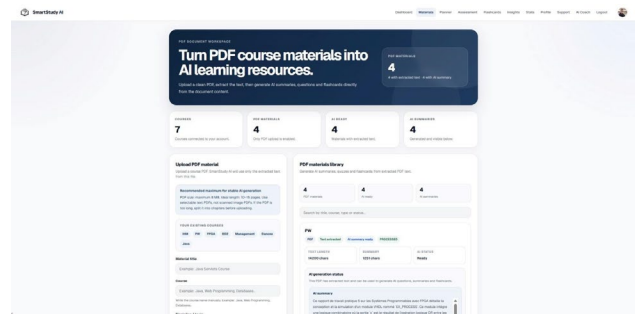


Fig. A3. Material import and PDF management area

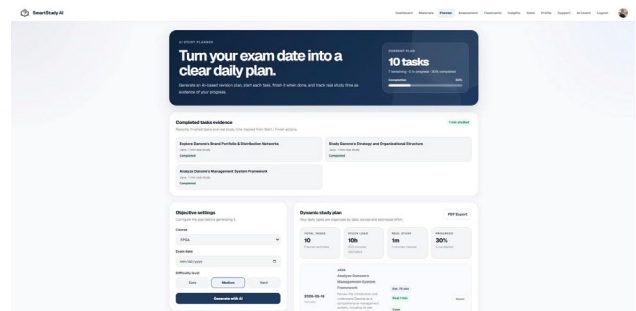


Fig. A4. AI-assisted study planner

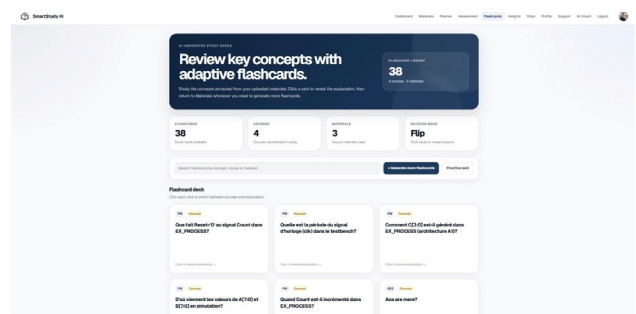


Fig. A5. Library of AI-generated flashcards

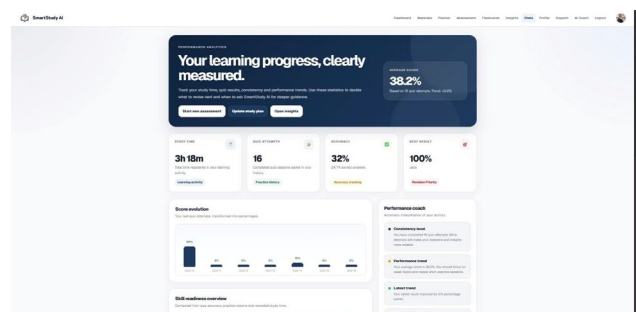


Fig. A6. Learning statistics and performance tracking