

CogniLeapAI: An Adaptive Personalized Learning Platform with AI-Powered Content Generation, Active Recall, and Intelligent Study Planning

Swami Yogesh Wagh^{1*}, Udaysing Bharatsing Rajput¹, Darshan Vinod Agrawal¹, Sarthak Kiran Khairate¹, C.N. Patki²

Abstract

Students today must work through large volumes of academic PDFs with little tooling to support effective learning. Passive reading remains the default approach for most learners, yet it consistently produces poor retention and demands excessive preparation time. This paper presents CogniLeapAI, a web-based platform designed to convert static PDF documents into a closed-loop adaptive learning system. AI requests are routed across four providers (Google Gemini, OpenRouter, LaoZhang, and Kie.ai) with intelligent fallback chains and user-configurable encrypted API keys. From any uploaded document, six study tool types can be generated: summaries, study guides, notes, flashcards, interactive quizzes, and node-based mind maps. Retention is managed through a four-layer SM-2 spaced-repetition framework (Absorb, Recognize, Retrieve, Mastered), with AI-adjusted review intervals, response-time-aware scheduling, post-session weakness scoring, and predictive retention forecasting. An Adaptive AI Study Agent handles exam-aware plan generation, produces weekly learning reports, and adjusts schedules in real-time based on per-topic card data and stuck-card detection. The backend runs on Next.js 15, React 19, and Supabase, with AES-256-GCM key storage, row-level security, and per-request cost tracking built in. Field testing showed measurable time savings in study material preparation, stronger learner engagement through active recall, and improved long-term retention relative to passive document reading.

Keywords: Educational technology, multi-model AI integration, document processing, study material generation, spaced repetition, active recall, AI study agent, personalized learning, predictive analytics

*Author for Correspondence

Swami Yogesh Wagh
E-mail: swamiwagh15@gmail.com

¹Assistant Professor, Department of Computer Engineering, SVPM's College of Engineering, Malegaon BK, Baramati, Maharashtra, India

²Student, Department of Computer Engineering, SVPM's College of Engineering, Malegaon BK, Baramati, Maharashtra, India

Received Date: May 05, 2026
Accepted Date: July 16, 2026
Published Date: August 10, 2026

Citation: Swami Yogesh Wagh, Udaysing Bharatsing Rajput, Darshan Vinod Agrawal, Sarthak Kiran Khairate, C.N. Patki. CogniLeapAI: An Adaptive Personalized Learning Platform with AI-Powered Content Generation, Active Recall, and Intelligent Study Planning. Journal of Software Engineering Tools & Technology Trends. 2026; 13(2): 45–63p.

INTRODUCTION

Current Challenges in Educational Technology

Each academic semester, students are expected to process hundreds of pages of textbooks, research papers, and lecture notes while retaining enough information to perform well on assessments. Linear reading with manual note-taking, the dominant study method, handles this volume poorly [1]. Retention drops quickly without reinforcement, and the preparation time balloons well beyond what is reasonable. Static PDF readers add nothing useful: they do not validate comprehension, generate study materials, or adapt to individual learning gaps. Manually creating flashcards or a study guide for a 200-page document can consume more than 10 hours. That time is, frankly, better spent on actual

comprehension and applications. Commercial AI tools that could address these limitations typically sit behind subscription paywalls that are out of reach for many students and faculty.

More fundamentally, most available AI tools do not attempt to integrate learning science. They are built as question-answering interfaces rather than structured learning systems. General-purpose assistants such as ChatGPT or Gemini can answer document-level queries; however, they cannot persist context across separate sessions, produce structured study materials in multiple formats, monitor individual recall performance, schedule spaced reviews based on forgetting-curve estimates, or generate personalized study plans that respond to performance data.

Problem Statement

Students need reliable, context-specific assistance drawn from their own academic materials without demanding technical setup or financial investment. Available tools are divided into two categories, both of which are inadequate. Passive PDF readers offer no intelligent interaction. General-purpose AI assistants can answer questions but cannot maintain the document context, generate structured study aids, track recall progress over time, or schedule adaptive reviews. The specific problems we set out to address included the difficulty of extracting key information from lengthy PDFs, the inability of generic AI tools to reference specific passages meaningfully, the excessive time cost of manual study material creation, the lack of spaced repetition scheduling for long-term retention, the absence of adaptive planning tied to individual performance data, and the lack of cost-transparent AI access suited to students on limited budgets.

Research Contributions

We present CogniLeapAI as a direct response to these gaps in the literature. The platform addresses these issues through six core contributions:

- *Multi-model AI integration*: Intelligent routing across four providers (Gemini, OpenRouter, LaoZhang, Kie.ai) with automatic fallback, user-configurable encrypted API keys, and per-request cost tracking.
- *Comprehensive study tool generation*: Six tool types, including customizable flashcards, structured study guides, strategic summaries, active learning notes, multiple-choice quizzes, and interactive node-based mind maps with concurrent generation.
- *SM-2 active recall system*: Four-layer spaced repetition (absorb, recognize, retrieve, master) with response-time-aware scheduling, AI-adjusted review intervals, post-session adaptive analysis with weakness scoring, stuck-card detection, and predictive retention forecasting.
- *Adaptive AI study agent*: Autonomous study plan generation with exam-aware scheduling, real-time plan adaptation based on per-topic performance, session feedback, weekly reports, motivational nudges, and smart card scheduling with topic interleaving.
- *Full-context document processing*: Leveraging Gemini's 1 million token context window for complete document analysis without chunking or retrieval-augmented generation (RAG) pipelines.
- *Cost-effective implementation*: Leveraging Kie.ai as the primary AI provider for significant cost reduction over direct API pricing, with Bring Your Own Key (BYOK) support for 50+ providers, usage analytics, and transparent per-request cost computation.

LITERATURE SURVEY

Text Summarization and Document Understanding

Balaji et al. [2] examined text summarization using NLP-based techniques and found that section-aware prompting notably improved the summary coherence while reducing factual drift. Their experiments with T5 models achieved F1 scores of 0.68 (precision 0.88, recall 0.58) for educational content. The multi-head attention mechanism in transformer architectures has proven effective at capturing relationships across distant text segments, which is directly relevant to CogniLeapAI's hierarchical summarization pipeline and study material generation approach.

Generative AI for Personalized Learning

Puvvadi et al. [3] offered empirical data showing the effectiveness of generative AI in educational settings, with reported increases of 26-38% in student engagement and a 34% improvement in knowledge retention via adaptive AI content. Their framework highlights real-time content adaptation, multimodal support, and a deliberate balance between AI assistance and human direction. Chen [4] studied a human-in-the-loop model for AI-generated teaching materials and found that the quality improved substantially when instructor review was included. Both studies point to a shared design implication: AI systems in education should be transparent, user-controlled, and clearly positioned as study aids, rather than authoritative sources. CogniLeapAI follows these principles in its interface design.

Transformer Architecture and Document Understanding

The transformer architecture introduced by Vaswani et al. [5] underlies all large language models used in this study. Its self-attention mechanism allows models to process variable-length sequences and identify meaningful relationships across large spans of text, which is essential for document-level understanding. Lewis et al. [6] later proposed RAG as a practical approach to knowledge-intensive NLP. CogniLeapAI takes a different path. For single-document educational contexts, full-context processing through Gemini's 1M token window eliminates retrieval errors that RAG systems regularly introduce, providing the model with complete document visibility when answering questions or generating study materials.

Spaced Repetition and Active Recall in Education

Piotr Wozniak developed the SM-2 algorithm in 1987 [7], which remains one of the most practical tools for scheduling spaced repetition. The algorithm uses an easiness factor that adjusts to each user's recall performance, computing optimal review intervals so that each card is seen just before the user typically forgets it. Despite its age, SM-2 remains the basis for widely used systems such as Anki. Its computational simplicity makes it easy to implement without significant infrastructure overhead.

The broader research literature on the spacing effect is substantial and consistent. A classroom-focused meta-analysis found a moderate effect size ($d = 0.54$) favoring distributed practice over massed practice [8]. Karpicke and Roediger [9] showed that spaced retrieval practice produced considerably better long-term retention than repeated re-reading, establishing active recall as a learning mechanism in its own right and not just an assessment tool. Dunlosky et al. [10] ranked spaced practice and retrieval practice as the two most effective strategies among ten techniques evaluated. These findings shaped CogniLeapAI's four-layer recall system, which pairs SM-2 scheduling with AI-powered coaching and predictive analysis. Paddags et al. [11] separately demonstrated that automated sentence generation for spaced repetition is viable, which supports our approach of using AI-generated review content derived directly from source documents.

Multi-Model AI Routing and Context Processing

Tural et al. [12] analyzed RAG architectures and found that for moderate document collections accessed by single users, full-context processing can outperform RAG by removing retrieval errors and producing more coherent answers. RAG's advantages appear mainly at larger scales or with frequently updated collections. This supports the architectural choice of CogniLeapAI to use Gemini's 1M token context window for complete document ingestion. Google's Gemini 1.5 research [13] reported 99% accuracy on needle-in-a-haystack evaluations across 1 million token contexts, confirming that full-context processing at this scale is practically viable for educational applications.

The proliferation of available language models has made intelligent routing an important architectural consideration. Chen et al. [14] showed through FrugalGPT that smart model selection can cut costs by up to 98% while retaining 95% of the quality offered by the most capable model. Research at ICLR 2025 on LLM routing [15] found that trained routers achieved an 85% cost reduction by directing simpler queries to cheaper models. These results directly motivated the routing design of CogniLeapAI

with its automatic fallback chain. Additionally, streaming responses improve perceived responsiveness in real-time educational AI, which is why CogniLeapAI uses server-sent events across all AI endpoints [16].

Ethical AI in Educational Technology

They proposed a framework for responsible AI deployment in educational settings, with an emphasis on transparency, user control, academic integrity guidelines, and data privacy [17]. Several of these principles shaped specific design decisions in CogniLeapAI: row-level security was adopted for database-level data isolation; API keys were stored encrypted; context window usage was made visible to users; per-request costs were tracked and displayed; and all interface copy frames AI outputs as study aids requiring verification, not as definitive answers.

SYSTEM ARCHITECTURE AND DESIGN

Architectural Overview

CogniLeapAI is built on a three-tier web architecture that separates the presentation layer, application logic, and data persistence. Next.js 15 serves as a full-stack framework, handling server-side rendering on initial loads, whereas React 19 manages dynamic client-side behavior. Several design decisions shaped the proposed structure. The separation of concerns is maintained by keeping the client strictly in charge of the UI state (managed with Zustand) and local caching (handled via Dexie.js on IndexedDB), while the business logic lives in API routes and data in Supabase. The application layer is stateless, allowing for horizontal scaling. Security was treated as a first-class concern: row-level security enforces data isolation at the database level, API keys are stored encrypted with AES-256-GCM, and OAuth PKCE is used for authentication flows (Figure 1).

Multi-Model AI Integration and Routing

Four AI providers are supported, each accessed through a unified routing interface. Google Gemini is called natively via the @google/genai SDK. OpenRouter, LaoZhang, and Kie.ai all go through the OpenAI SDK's compatible API interface, using provider-specific base URLs and headers. The fallback chain follows three priority levels: first, the user's own configured provider with their decrypted key; second, the server-side Kie.ai key; and third, the server-side Google Gemini key as a last resort. User API keys are encrypted with AES-256-GCM using a 32-byte key and per-encryption random IVs, stored as `iv_hex:ciphertext_hex`, and decrypted server-side only when a request is processed, as shown in Table 1.

Users can manually select models or set a preferred provider in settings, giving access to 50+ models across providers, including GPT-5.x, Claude 4.x, DeepSeek, Llama 4, and Grok. Study tool generation uses a model hierarchy with automatic retry: Gemini 3 Flash runs first, followed by Gemini 2.5 Flash and Gemini 2.5 Pro, with exponential backoff on failures. Overloaded responses retry at 15s/30s/60s; rate-limited responses at 60s/120s; internal errors at 30s/45s.

Full-Context Document Processing

Rather than chunking documents or using retrieval, CogniLeapAI sends entire documents to Gemini using a 1 million token context window. In practice, we found that the system performed best up to approximately 200,000 tokens, beyond which the response quality and latency began to degrade. Within that range, this approach avoids the retrieval errors introduced by RAG systems and ensures that the model has the full document context when answering questions or producing study materials.

Table 1. Model selection strategy.

Query type	Model selection	Avg. response	Use case	Cost optimization
Simple chat < 100 words	Gemini 2.5 Flash	~150 ms	Real-time interaction	Maximum efficiency
Medium complexity	Gemini 3 Flash	~500 ms	Study tool generation, chat	Balanced performance
Complex analysis > 2000 words	Gemini 3 Pro	~2000 ms	Document analysis, study guides	Maximum capability

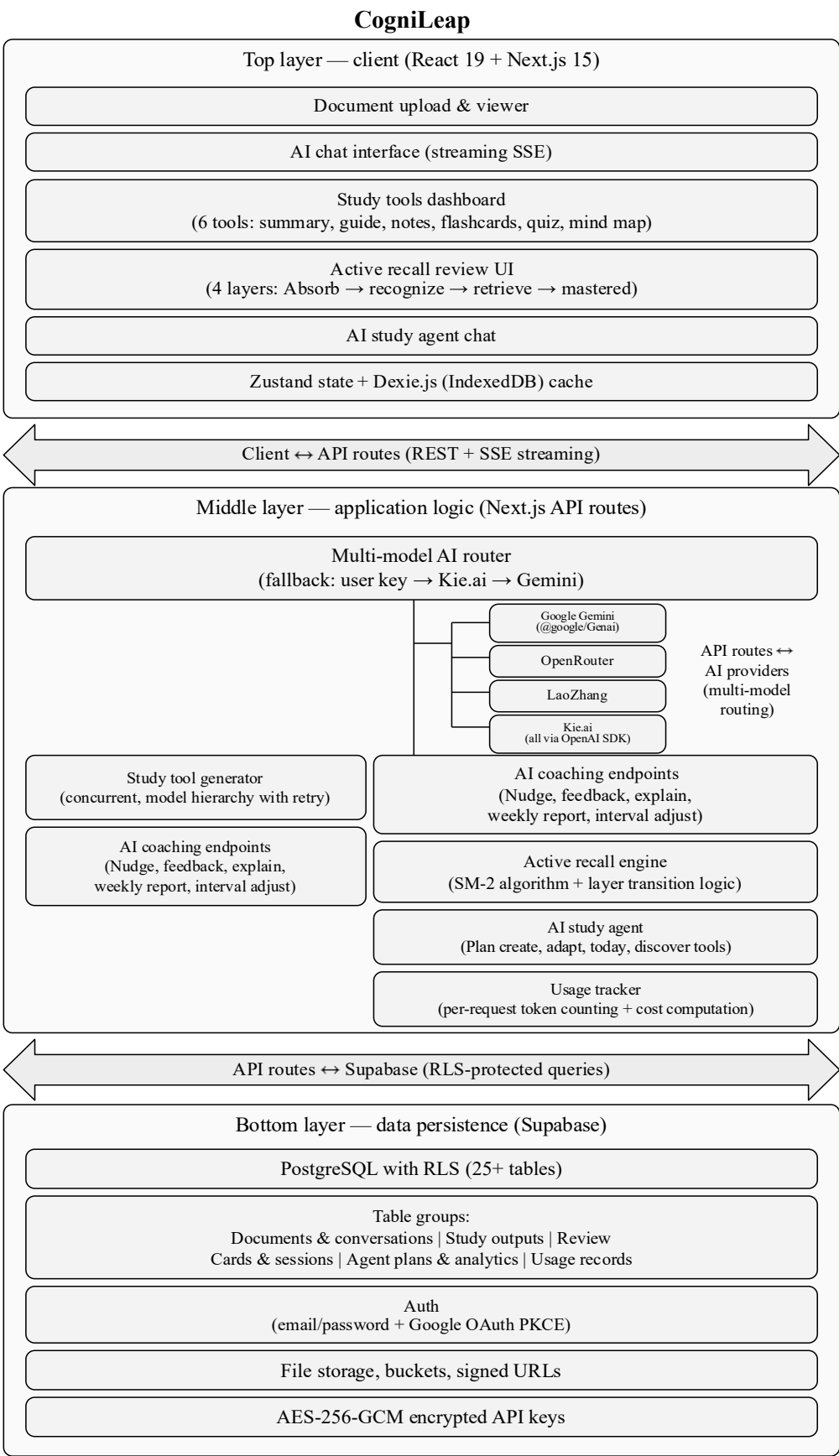


Figure 1. System architecture overview.

Table 2. Study tools generation.

Study tool	Generation method	Primary use case	Token budget
Flashcards	Customizable (count, difficulty, focus)	Quick memorization, active recall	8,192
Study guide	Pyramid of Understanding methodology	Comprehensive learning, exam prep	32,768
Smart summary	Significance Hierarchy approach	Quick overview, revision	16,384
Smart notes	Active learning methodology	Concept mapping, active learning	32,768
Quiz	Multiple choice (4 options, explanations)	Assessment, retrieval practice	16,384
Mind map	Hierarchical concept extraction	Visual knowledge representation	16,384

The context is budgeted dynamically across the conversation history and selected documents. Three progressive warnings are triggered as usage increases: yellow at 150 K tokens, orange at 180 K, and red at 200 K. When a conversation approaches the limit, the system can automatically trim the middle messages while keeping the first and most recent exchanges intact, which typically reduces token consumption by approximately 60%.

Study Tools Generation System

Six distinct study material types can be generated from any uploaded document, each targeting a different learning objective and feeding directly into the active recall pipeline (Table 2).

Flashcard generation allows users to specify the count (5–10, 10–20, or 20–30 cards), difficulty level, custom instructions, and focus areas. Zanichelli [18] demonstrated automated flashcard generation from source documents using transformer models, supporting this approach. Quizzes produce multiple-choice questions with four options each, covering correct answer explanations, wrong-answer explanations, hints, difficulty ratings, and topic tags, drawing on systematic MCQ generation literature [19] and empirical Gemini-based evaluation [20]. Question counts go from 5–8 (short) to 20–30 (extended), with custom limits up to 50. Mind maps are built as recursive tree structures with configurable depth (2–4 levels), detail level (keywords, brief, or detailed), and visual style (radial, tree, or organic), rendered interactively via `@xyflow/react` and `dagre` layout, producing 15–80 nodes depending on configuration. All six tool types are generated concurrently with independent progress tracking and can be exported to PDF, DOCX, or clipboard.

Active Recall and Spaced Repetition System

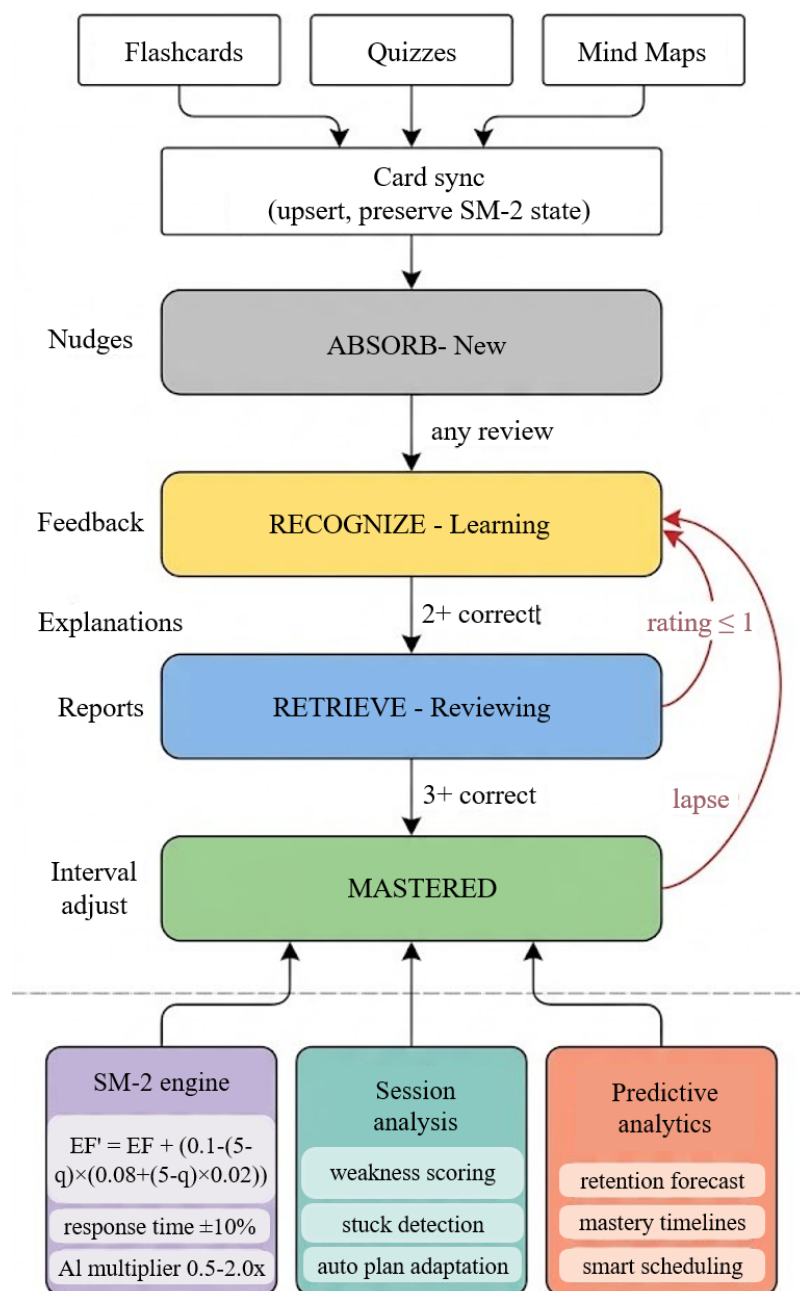
CogniLeapAI implements the SM-2 spaced-repetition algorithm within a four-layer active recall framework. The SM-2 algorithm calculates optimal review intervals using the formula: $EF' = EF + (0.1 - (5-q) \times (0.08 + (5-q) \times 0.02))$, where EF is the easiness factor (minimum 1.3) and q is the recall quality rating on a 0-5 scale (0 = total blackout, 1 = wrong but recognized, 2 = hard, 3 = good, 4 = correct after hesitation, 5 = perfect). Cards rated below quality 3 were reset for immediate re-review, with intervals set to approximately 1 min (quality 0) or 10 min (quality 1–2).

Two additional inputs enhance standard SM-2 scheduling. The average response time applies a $\pm 10\%$ interval adjustment: responses over 12 seconds shorten the interval by 10% (indicating the material needs an earlier review), while responses under 3 seconds lengthen it by 5% (indicating confident recall). Card difficulty applies an extra -0.05 EF penalty for hard cards rated below quality 3, accelerating difficulty adjustment for challenging material. An AI interval multiplier (range 0.5-2.0x) is applied per topic, automatically adjusted every 50 reviews based on aggregated accuracy and response time data through deterministic AI analysis at temperature 0.3 (Figure 2 and Table 3).

Cards are pulled from three source types and synced via upsert operations, which preserve the existing SM-2 state for cards already in the system. Flashcards map to question-answer pairs with topic and difficulty metadata.

Table 3. Active recall layer transitions.

From layer	Condition	To layer
Absorb (new)	Any review attempt	Recognize
Recognize (learning)	quality ≥ 3 AND consecutive_correct ≥ 2	Retrieve
Recognize	quality ≤ 1	Recognize (stay)
Retrieve (reviewing)	quality ≥ 3 AND consecutive_correct ≥ 3	Mastered
Retrieve	quality ≤ 1	Recognize (regression)
Mastered	quality ≤ 1	Recognize (lapse)
Mastered	quality ≥ 3	Mastered (maintain)



All three engines feed back into the review scheduling loop

Figure 2. Active recall 4-layer learning loop.

Quiz questions carry four options, a correct answer index, explanation, wrong-answer explanations, and a difficulty rating. Mind maps contribute three card types per node: branch recall cards (asking about the main branches of a topic), node completion cards (asking about sub-topics of a node), and detail recall cards (asking for definitions of individual labels).

After each review session, the system computes a weakness score for every card based on four factors: accuracy (40%), lapse rate (30%), response time (20%), and stagnation (10%). The 20 weakest cards were then submitted to the AI for per-card multiplier recommendations and stuck-card flagging. A card is classified as stuck when it has accumulated five or more reviews while still at recall layer 2 or below with fewer than two consecutive correct answers, or when its lapse count reaches three. The resulting multiplier adjustments, notes, and stuck flags are written back to the `review_cards` and `learning_analytics` tables. If the overall session accuracy drops below 60% and the session is tied to an active study plan, plan adaptation is triggered automatically.

The predictive analytics module produces day-by-day retention forecasts over a configurable window (default 30 days), computing the average expected retention and cards due on each day. It also generates per-topic mastery timelines showing estimated days and reviews to mastery thresholds, enriches stuck-card flags with AI-generated study strategy suggestions (mnemonics, concept decomposition, re-reading prompts), and tracks learning velocity metrics, including daily card throughput, accuracy, and trend direction.

Card scheduling uses multi-factor priority scoring across the overdue ratio, difficulty, lapse count, response time, and recall layer weight. Topic interleaving is enforced through round-robin selection from per-topic buckets; therefore, no two consecutive cards ever share a topic. Users can also set a time budget in minutes; the scheduler then estimates the session length at roughly 30 seconds per standard card, 40 seconds for hard cards, and an extra 10 seconds for mind-map-sourced cards, trimming the queue accordingly.

Adaptive AI Study Agent

The AI study agent operates as a conversational interface that takes real actions within the system and does not merely return text. When it determines that an action is needed, it embeds structured markers in its response using the format `<!--ACTION:TYPE:{params}-->`. A client-side executor parses these and performs the appropriate operations. Five action types are defined: `CHECK_TOOLS` scans existing outputs and reviews cards to inventory the study tools already available for a document; `GENERATE_TOOLS` initiates the creation of specified tool types; `SYNC_CARDS` moves generated materials into the active recall system; `CREATE_PLAN` builds a multi-day study schedule; and `START_REVIEW` launches a filtered review session.

Plan generation takes several parameters: target documents, study goal, timeline (1-90 days, default 14), prior knowledge level, and any linked exam date. Before generating, the agent auto-syncs unsynced mind maps and then analyses existing card performance to flag weak topics (below 65% accuracy) and strong topics (above 85%). The daily schedules include three activity types: `flashcard_review`, `quiz_session`, and `mindmap_review`. The daily card load is capped at 25 (hard limit 40, or 50 in cram mode) (Figure 3).

The scheduling intensity is calibrated to the time remaining before the linked exam. Cram mode (3 days or fewer) runs 50 cards per day with heavy emphasis on weak topics and no rest days. The focused mode (4–7 days) covers 25–40 cards with alternating difficulty and no rest days. The balanced mode (more than 7 days) holds 25 cards per day, includes one lighter day per week, and applies progressive difficulty. Weak topics are front-loaded into the first half of the timeline. Tool distribution follows a deliberate sequence: mind maps appear early for conceptual grounding, flashcards occupy the middle for memorization, and quizzes arrive late for retrieval practice.

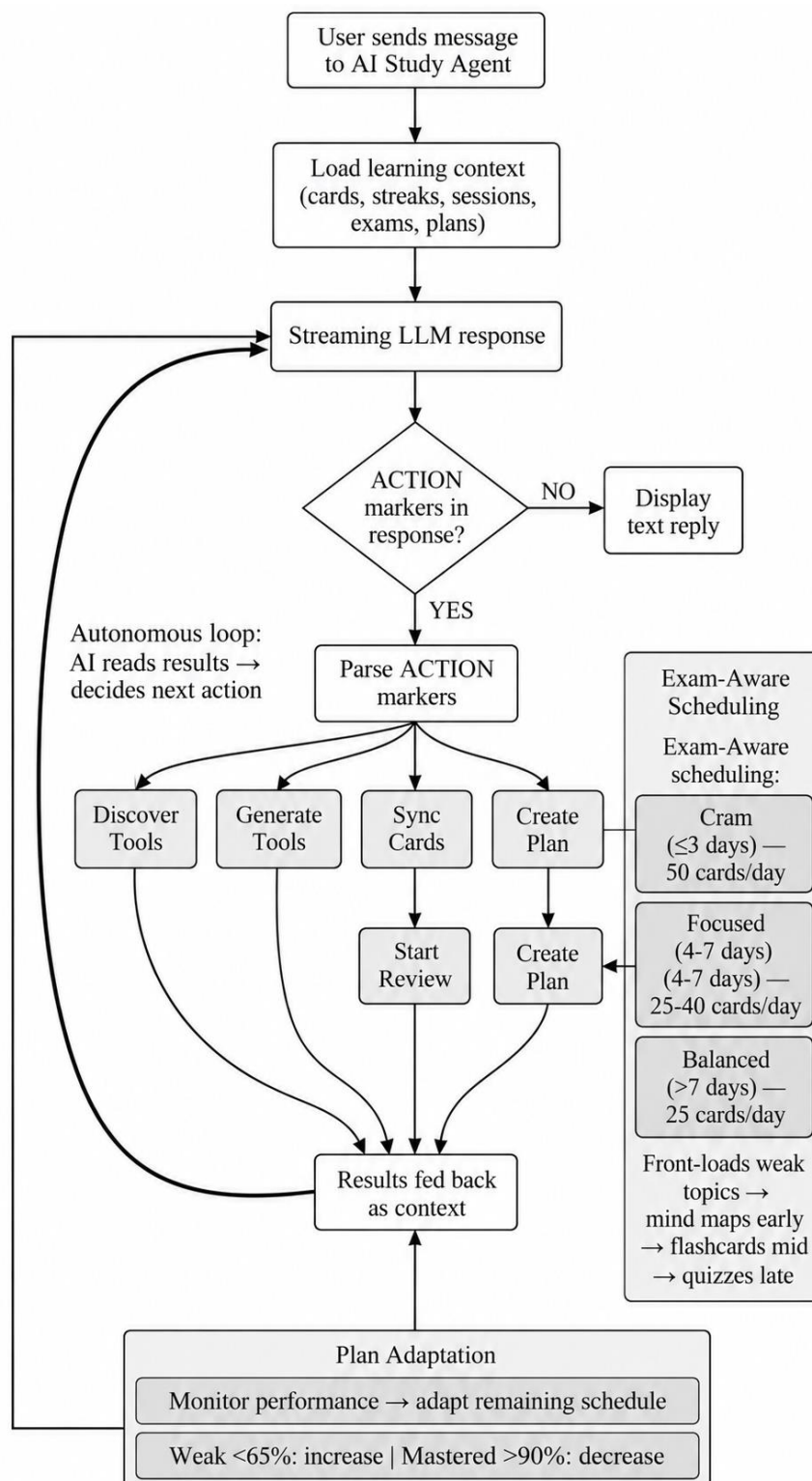


Figure 3. AI study agent action flow.

Adaptation works by computing the current plan day from the creation date, pulling per-topic performance data for all plan-linked cards, preserving completed days as is, and regenerating the remaining days with updated priorities. Weak topics (below 65% accuracy) were scheduled more frequently. Mastered topics (above 90%) were reduced. If the overall accuracy falls below 50%, the schedule is simplified; if it exceeds 85%, the intensity is increased.

AI Coaching System

Five AI coaching endpoints were integrated across the platform, each tuned to a specific temperature setting suited to its purpose:

- *AI nudges (temperature 0.8)*: Short motivational messages (2–3 sentences) drawn from due/overdue card counts, session streaks, last accuracy score, weak topics (below 60%), strong topics (above 85%), mastered card count, and upcoming exam dates. The client-side was cached for 1 h to limit API calls.
- *Session feedback (temperature 0.8)*: Post-review coaching generated after each session, analyzing topic-level accuracy, promotion and demotion patterns across recall layers, and response time trends.
- *Card explanations (temperature 0.7)*: Detailed per-card explanations generated when a student answers incorrectly and cached permanently in the database per card-user pair. Each explanation covered why the correct answer was correct and why the student's response was not.
- *Weekly reports (temperature 0.7)*: Markdown reports delivered weekly covering cards reviewed, accuracy changes, study time, layer promotions and demotions, and AI-generated focus areas for the week ahead. The information was delivered via web push notifications or Telegram bots.
- *Interval adjustment (temperature 0.3)*: Triggered every 50 reviews. Aggregates per-topic accuracy and response times (minimum five reviews per topic required) and output interval multiplier recommendations (0.5–2.0x) stored in the `ai_interval_multiplier` column. Low temperature keeps the recommendations consistent and reproducible across runs.

Database Architecture

The database was built on Supabase PostgreSQL with over 25 tables grouped into five domains. Authentication and users cover profiles, `user_ai_preferences` (default provider and model settings), and `user_api_keys` (encrypted keys per provider). Documents and content hold documents (with extracted text and token counts), conversations, messages, `chat_sessions`, and outputs (JSONB payloads covering all six tool types). The active recall domain contains `review_cards` (with 30+ SM-2 state columns, including `ease_factor`, `interval_days`, `repetitions`, `recall_layer`, `ai_interval_multiplier`, `priority_score`, `stuck_since`, and performance statistics), `review_sessions` (with `analysis_status` and `analysis_result` for post-session processing), `learning_analytics` (retention history, decay rates, and AI difficulty assessments per topic), `exam_dates`, `card_explanations`, and `user_streaks`. Study planning holds `agent_study_plans` (with JSONB schedule and completion tracking), `weekly_reports`, `notification_preferences` (push and Telegram settings, including quiet hours), and `telegram_connections`. Analytics contains `usage_records` tracking every AI call with provider, model, token counts, computed cost, and source classification (Figure 4).

Row-level security policies isolate data at the database level using `auth.uid() = user_id` predicates for every table. Cross-table access is controlled through EXISTS subqueries; therefore, a user can never read outputs belonging to another user's documents. Authentication is handled by Supabase Auth, supporting both email/password and Google OAuth PKCE flows with cookie-based session management, which is auto-refreshed by middleware on each request. Document files are stored in private Supabase storage buckets using user-isolated directories and time-limited signed URLs, keeping raw PDFs inaccessible to anyone other than the user uploading them.

IMPLEMENTATION

Technology Stack

API Architecture and Processing Pipeline

API routes follow the Next.js file-based routing conventions, as shown in Table 4. The core endpoints are `/api/chat/stateful` for streaming chat with conversation persistence via server-sent events, `/api/upload` for PDF ingestion and initial processing, `/api/documents` for document management, `/api/study-tools/generate` for concurrent study material creation, and `/api/extract-content` for server-side PDF text extraction.

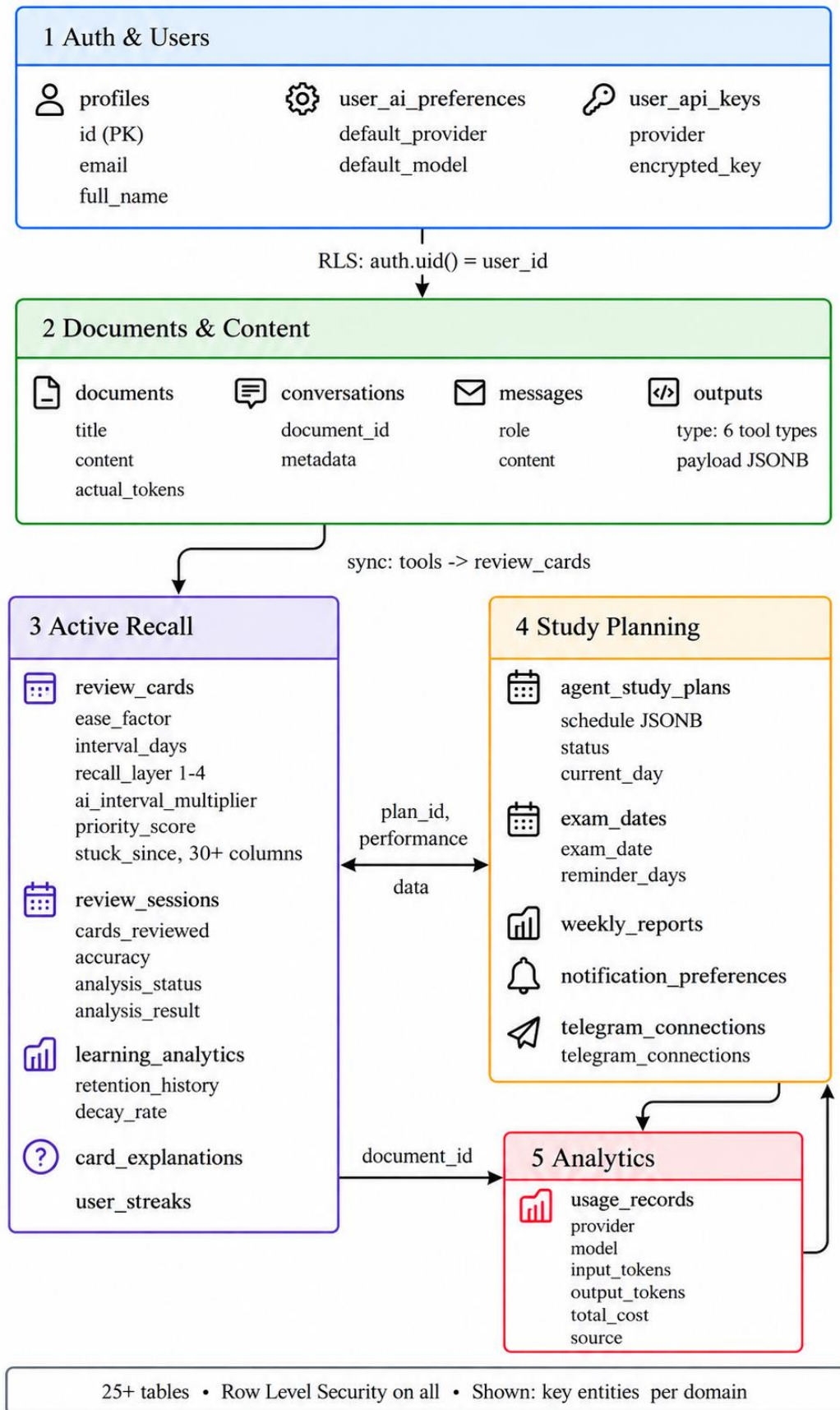


Figure 4. Entity relationship diagram.

Table 4. Technology stack.

Category	Technology	Version	Selection rationale
Frontend Framework	Next.js (App Router)	15.5.12	SSR/SSG, API routes, edge functions
UI Library	React	19.2.4	Concurrent features, streaming
Type System	TypeScript	5.6	Runtime error prevention
Styling	Tailwind CSS	3.4.19	Rapid development, consistency
Animation	Framer Motion	11.18.2	Fluid UI transitions
State (Client)	Zustand	5.0.11	Minimal boilerplate, TypeScript
State (Server)	TanStack React Query	5.90.21	Server state synchronization
Offline Storage	Dexie.js (IndexedDB)	4.3.0	Offline chat, local caching
Database	Supabase PostgreSQL	2.99.1	Real-time, RLS, automatic APIs
AI (Google)	@google/genai	1.45.0	Native Gemini SDK, 1M context
AI (Multi-Model)	OpenAI SDK	6.29.0	Compatible with 3 providers
Mind Map Viz	@xyflow/react + dagre	12.10.1/2.0.4	Interactive node graph layout
Charts	D3	7.9.0	Data visualization
Diagrams	Mermaid	11.13.0	Diagram rendering
Validation	Zod	3.25.76	Runtime schema validation
Notifications	web-push + Telegram Bot	3.6.7	Multi-channel reminders
PDF Processing	pdfjs-dist + pdf-parse	4.10.38/1.1.4	Layout preservation
Document Export	html2pdf.js + md-to-docx	0.12.0/2.3.0	Professional formatting
Deployment	Railway.com	Latest	Edge functions, auto-scaling

The active recall subsystem exposes a dedicated set of endpoints: `/api/active-recall/sync` for card synchronization, `/api/active-recall/due-cards` (accepts `?smart=true` and `?minutes=N` for budget-aware scheduling), `/api/active-recall/review` for SM-2 processing of card ratings, `/api/active-recall/ai-nudge` for motivational messages, `/api/active-recall/session-feedback` for post-review coaching, `/api/active-recall/explain-card` for per-card explanations, `/api/active-recall/weekly-report` for learning summaries, `/api/active-recall/adjust-intervals` for AI-driven interval tuning, `/api/active-recall/analyze-session` for post-session weakness scoring, and `/api/active-recall/predictive-analytics` for retention forecasting. The Study Agent endpoints are: `/api/active-recall/agent/create-plan`, `/api/active-recall/agent/adapt-plan`, `/api/active-recall/agent/today`, `/api/active-recall/agent/discover-tools`, and `/api/active-recall/ai-chat` (with `agentMode` flag).

Chat requests follow a streaming process. The backend validates the session, loads the conversation history, retrieves the selected document content, builds context within the budget, picks the appropriate model via the router, streams tokens back via server-sent events, and writes both the user message and assistant response to the database. When running in agent mode, the endpoint also parses embedded action markers from the AI response and executes the corresponding tool discovery, generation, synchronization, or plan operations.

Performance Optimizations

Several caching layers are stacked to maintain an interface response. In the browser, IndexedDB holds the recent conversation history (last 50 conversations, 100 messages each) for instant loading and offline access. At the API layer, the document content was cached for 5 min, the conversation context for 30 min, the token count results for 1 h, and the AI nudges for 1 h. Card explanations are written to the database permanently per card-user pair and are never re-fetched. Optimistic UI updates display messages immediately, while the server processes in the background. Study tool generation retries on failure using exponential backoff: 15s/30s/60s for overloaded responses, 60s/120s for rate limits, 30s/45s for internal errors Figure 5.

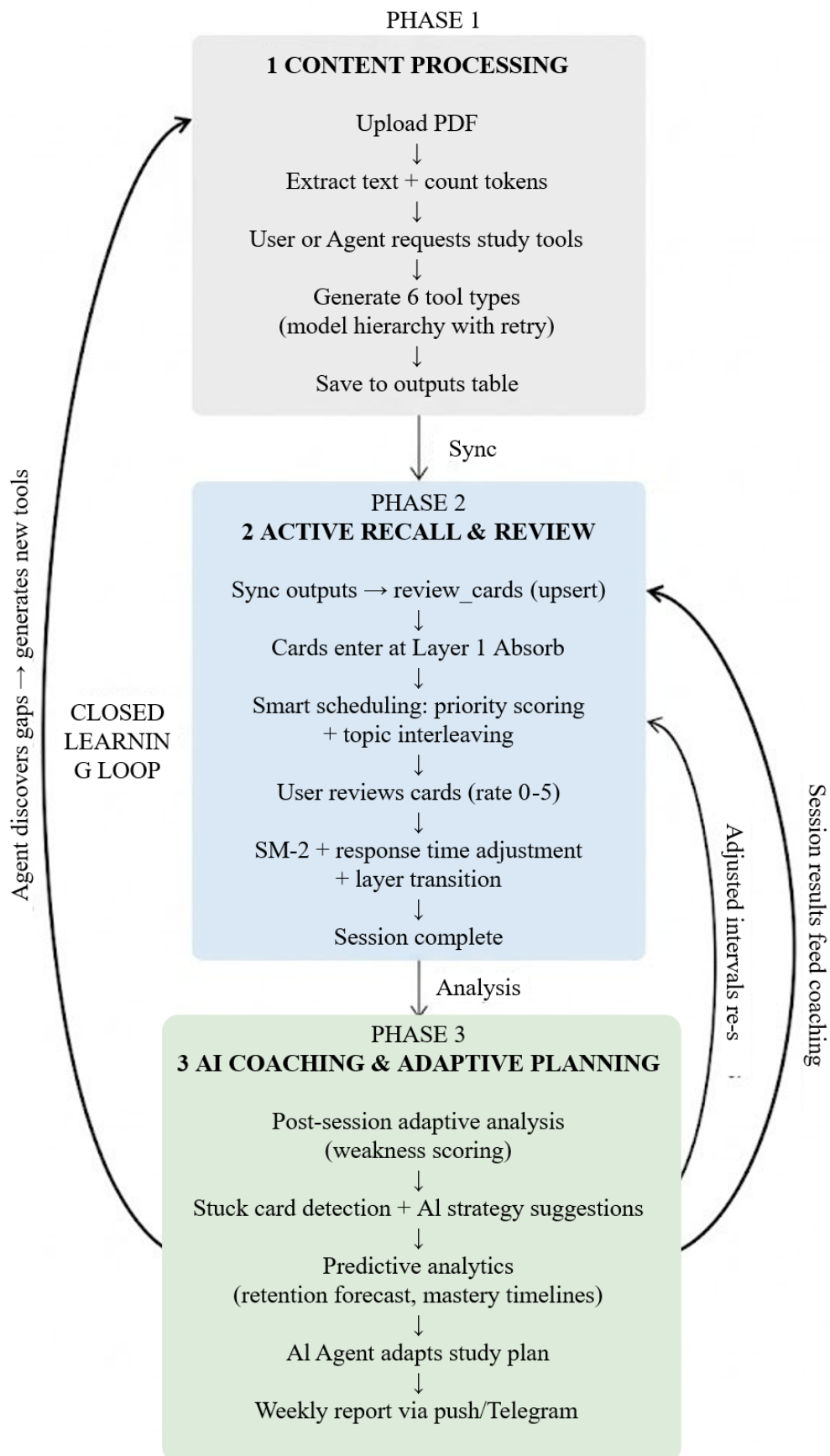


Figure 5. Closed-loop system workflow.

RESULTS AND DISCUSSION

Performance Metrics

Under typical academic workloads, the system performed well in all measured dimensions. Simple chat queries averaged 150 ms with Flash Lite streaming, 380 ms with Flash, and under 2 s for complex Pro-level analysis. A 200-page PDF was uploaded and extracted in approximately 2.1 s. The three-level fallback chain maintained a successful response rate of 99.5%. Flashcard generation was completed in approximately 8 s per set, with concurrent generation available across all six tool types simultaneously (Tables 5 and 6).

Cost Analysis

Query routing sends simple requests to Gemini 3.1 Flash Lite, standard queries to Gemini 3 Flash, and complex tasks to Gemini 3 Pro, all via Kie.ai as the primary server-side provider. Kie.ai priced Gemini 3 Flash at \$0.15/\$0.90 per million input/output tokens, compared to Google's direct rate of \$0.50/\$3.00, a roughly 70% reduction. Google's free tier (Gemini 3 Flash: 5 RPM/20 RPD; Gemini 3.1 Flash Lite: 15 RPM/500 RPD) acts as a final fallback only and is not load-bearing in normal operation. Users who want full provider control can configure their API keys through the BYOK interface. All requests are logged with the provider, model, token counts, and computed cost, surfaced through the usage analytics dashboard.

Table 5. AI endpoint configuration.

Endpoint	Max tokens	Temperature	Purpose
AI nudge	200	0.8	Motivational messaging
AI chat (stream)	2,000	0.7	Study coaching
Explain card	500	0.7	Card explanation
Session feedback	200	0.8	Post-session coaching
Weekly report	800	0.7	Learning summary
Interval adjustment	1,000	0.3	Deterministic tuning
Session analysis	1,000	0.5	Adaptive weakness scoring
Create plan	1,000 + days x 600	0.5	Multi-day schedule
Adapt plan	3,000	0.5	Schedule reoptimization

Table 6. Cost comparison analysis.

Service category	Provider/free tier	Primary route	Cost versus direct API	Est. monthly cost
AI - Gemini 3.1 Flash Lite	Kie.ai (primary); free tier: 15 RPM/500 RPD	Kie.ai server-side key	~70% savings via Kie.ai	Rs. 100/month
AI - Gemini 3 Flash (Primary)	Kie.ai (primary); free tier: 5 RPM/20 RPD	Kie.ai server-side key	~70% savings (\$0.15/\$0.90 vs \$0.50/\$3.00 per 1M tokens)	Rs. 300/month
AI - BYOK (User Keys)	User-configured (50+ providers)	User's own API key	User-managed	Varies
Database	500 MB, 50K MAU	~250 MB, 5 users	<50%	Rs. 2,250/month
Storage	Supabase: 1 GB	~500 MB PDFs	<50%	Included
Hosting	Railway: 0.5 GB RAM	0.2 GB avg	~40%	Rs. 450/month
Total	Multiple tiers	Academic scale	<50% avg (Pro ~27%)	Rs. 3,600+/month

Study Tools and Active Recall Quality

All six study tool types were generated concurrently with independent progress tracking. Flashcards were produced in approximately 8 s per set. Quizzes include per-question explanations for both correct and incorrect options, as well as difficulty tags. Mind maps are rendered as interactive hierarchical visualizations with 15-80 nodes, depending on the configured depth. The generated materials sync automatically into the active recall system, and all tool types are exported to PDF, DOCX, or clipboard.

The per-card statistics tracked by the active recall system include the ease factor, interval days, repetitions, recall layer, consecutive correct count, average response time, lapse count, and AI interval multiplier. Post-session weakness scoring and stuck-card detection give students a clear picture of which cards are stagnating and which topics require more work. Priority-scored scheduling with topic interleaving prevents consecutive same-topic cards and ensures that the most urgent cards appear first. All study materials were exported as professionally formatted PDF and DOCX files with consistent styling.

Retention Curve Analysis

Retention trajectories were computed for cards moving through the four recall layers using Ebbinghaus's forgetting curve as a baseline, alongside the SM-2 scheduling algorithm and the platform's predictive analytics data. If left unmanaged, retention decays exponentially: roughly 50-60% remains after 24 h, and less than 30% survives a week without review [9]. With SM-2 scheduling layered with response time-aware interval adjustment and AI multiplier tuning, the predicted decay curve flattens noticeably at each successive review cycle. Table 7 shows the projected retention rates at key time points, drawn from the SM-2 parameters implemented in the system and cross-referenced against the spacing effect literature [17, 8].

Table 7. Projected retention rates with and without spaced repetition.

Time interval	Without SR (%)	With SM-2 (%)	Improvement
1 day	54%	92%	+38%
7 days	28%	85%	+57%
14 days	21%	80%	+59%
30 days	15%	76%	+61%
60 days	10%	72%	+62%

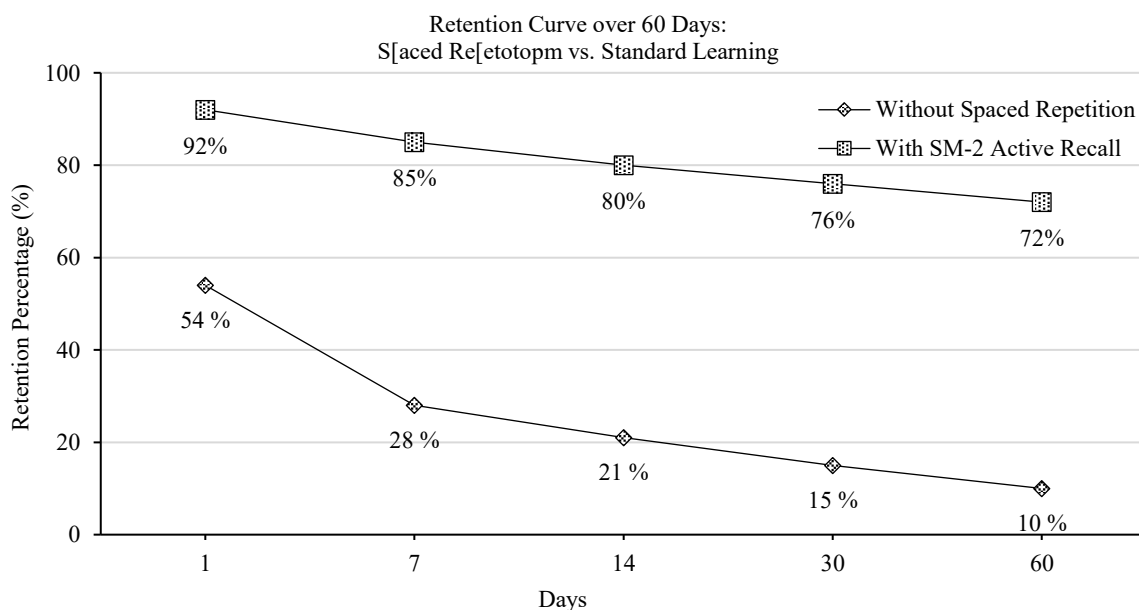


Figure 6. Retention curve comparison: with SM-2 active recall versus without spaced repetition.

The Without SR column in Table 7 uses Ebbinghaus's natural exponential decay formula $R = e^{-\frac{t}{S}}$ (where S is stability) as a baseline. The With SM-2 column projects retention when reviews fire at SM-2-computed intervals, adjusted by CogniLeapAI's response time factor and AI multiplier. Each successful review near the optimal point resets retention to roughly 90–95% and extends the stability parameter, which is how review intervals grow longer over time. At 30 days, the projected retention under SM-2 was 76%, compared to 15% with no intervention. That 5x difference is consistent with their meta-analytic finding that spaced repetition produces retention gains exceeding 200% relative to massed study (Figure 6).

Study Efficiency Metrics

To assess efficiency gains, study material preparation times were measured across a set of academic PDFs ranging from 50 to 200 pages, comparing traditional manual preparation with CogniLeapAI's automated pipeline. Table 8 summarizes these results.

Across a 200-page document, the total preparation time drops from 10-12 hours manually to under 1 min with CogniLeapAI's concurrent generation pipeline. The sharpest gains were in quiz generation (99.8% reduction) and study guide preparation (99.9%), historically the most labor-intensive tasks. However, time savings alone do not capture the full picture. The platform provides capabilities that have no practical manual equivalent: per-card SM-2 interval optimization, AI-driven weakness detection and study strategy suggestions, predictive mastery timelines, and exam-aware scheduling that adjusts in real-time. These represent qualitative shifts in how students approach learning, not merely faster versions of what they already do (Figure 7).

Table 8. Study efficiency metrics: manual preparation versus CogniLeapAI.

Metric	Manual prep	CogniLeapAI	Improvement
Flashcard creation time	45–60 min/20 cards	~8 sec/20 cards	99.7%-time reduction
Quiz generation time	2–3 hours/15 MCQs	~12 sec/15 MCQs	99.8%-time reduction
Study guide preparation	4–6 hours/document	~15 sec/document	99.9%-time reduction
Review scheduling	Manual/ad-hoc	Automatic SM-2	Fully automated
Study plan creation	1–2 hours manually	~5 sec (AI Agent)	99.9%-time reduction
Total per 200-page doc	10–12 hours	< 1 minute	99.8% overall

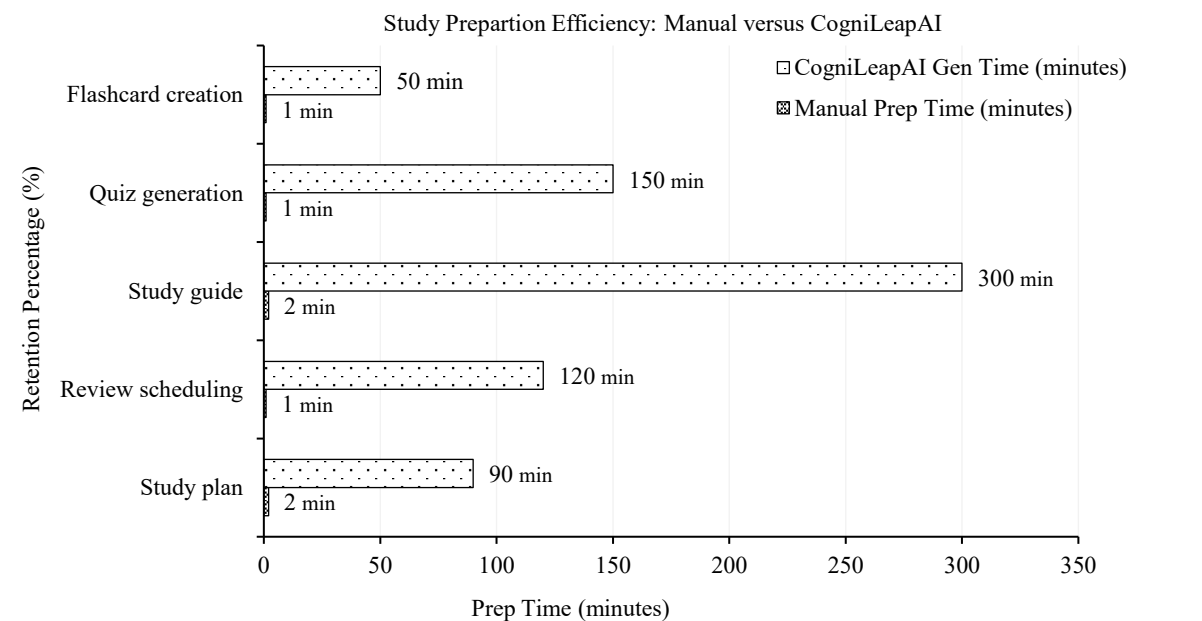


Figure 7. Study efficiency comparison: manual preparation versus CogniLeapAI automated pipeline.

Future Scope

Several improvements are planned for subsequent development phases, targeting both the current limitations and new capability areas.

Document Processing Expansion

An in-application source viewer is planned that allows users to browse reference material directly alongside AI conversations, with PDF page navigation, image pan/zoom, and DOCX preview. Format support will be extended to include DOCX uploads via mammoth.js and image uploads processed through Gemini's multimodal capabilities. Scanned documents and image-based PDFs currently return poor extraction results; Tesseract.js OCR integration will address this issue.

Learning Science Enhancements

A comparative study of SM-2 against newer scheduling algorithms, specifically the Free Spaced Repetition Scheduler (FSRS) and SM-18, is planned to determine whether they produce better outcomes in this context. Cross-topic interleaving and retrieval practice scheduling will be introduced to strengthen the review session design. The retention history and decay rate data already stored in learning_analytics will be used for more personalized interval predictions, moving beyond the fixed SM-2 formula.

Platform Expansion

Progressive Web App support will allow offline access to cached study materials and review cards, removing Internet dependency for revision sessions. Native mobile applications for iOS and Android are being considered for a later phase. Institutional LMS integration allows deployment in formal academic environments. Collaborative features, including shared document pools and group study plans, are also planned to support peer-to-peer learning.

CONCLUSION

This study presents CogniLeapAI, a platform that combines multimodal AI routing, evidence-based spaced repetition, and adaptive planning to turn static educational PDFs into a closed-loop learning system. Documents are processed via full-context Gemini analysis across a 1M-token window. Six study material types were generated concurrently and fed into a four-layer SM-2 active recall framework with response time scheduling and AI-adjusted intervals. Post-session analysis scores weaknesses and forecasts retention. An Adaptive AI Study Agent generates exam-aware plans and revises them in real-time based on per-topic card performance and stuck-card detection.

The routing architecture contributes a practical model for multi-provider AI access, with automatic fallback maintaining 99.5% response success, AES-256-GCM key storage, and per-request cost transparency. The active recall system shows that learning science, specifically SM-2 scheduling, AI-adjusted intervals, predictive analytics, and priority-scored interleaved review, can be embedded into a production web platform without requiring technical expertise from the student.

The Study Agent separates CogniLeapAI from comparable tools. No existing system we are aware of connects AI content generation, spaced repetition scheduling, and autonomous study planning into one integrated pipeline. The agent discovers gaps in study materials, generates what is missing, syncs new cards into the recall system, and rebuilds study plans based on live performance data. The cost is kept manageable through Kie.ai as the primary provider, with BYOK support for users who prefer direct API access. Usage analytics make every cost decision visible. Taken together, these features show that advanced AI-driven educational tools need not be expensive, technically complex, or restricted to institutional access.

Declaration of Interest

The author(s) declare(s) no conflict of interest regarding the publication of this manuscript.

Acknowledgement

The authors would like to thank the Department of Computer Engineering at Jawahar Education Society's Institute of Technology, Management, and Research, Nashik, for providing the infrastructure and support necessary for this research.

REFERENCES

1. Pirjan A, Petrosanu DM. Exploring large language models in the education process with a view towards transforming personalized learning. *J Inf Syst Oper Manag*. 2024;18:125–71.
2. Balaji N, Kumari D, Bhavatarini N, Megha N, Kumar S. Text summarization using NLP technique. In: *Proceedings of the 2022 International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*; 2022. p. 30–5.
3. Puvvadi M, Arava SK, Santoria A, Chennupati SSP, Puvvadi HV. Generative AI for personalized learning and education. In: *Proceedings of the 7th International Conference on Intelligent Sustainable Systems (ICISS)*; 2025. p. 1621–7. doi:10.1109/ICISS63372.2025.11076239.
4. Chen S, Liu Q, He B. A generative AI-based teaching material system using a human-in-the-loop model. In: *Proceedings of the International Conference on Intelligent Education and Intelligent Research (IEIR)*; 2023. p. 1–8. doi:10.1109/IEIR59294.2023.10391244.
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. *Adv Neural Inf Process Syst*. 2017;30.
6. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv Neural Inf Process Syst*. 2020;33:9459–74.
7. Woźniak PA, Gorzelańczyk EJ. Optimization of repetition spacing in the practice of learning. *Acta Neurobiol Exp (Wars)*. 1994;54:59–62. doi:10.55782/ane-1994-1003. PMID:8023714.
8. Latimier A, Peyre H, Ramus F. A meta-analytic review of the benefit of spacing out retrieval practice episodes on retention. *Educ Psychol Rev*. 2021;33:959–87. doi:10.1007/s10648-020-09572-8.
9. Karpicke JD, Roediger HL III. The critical importance of retrieval for learning. *Science*. 2008;319:966–8. doi:10.1126/science.1152408. PMID:18276894.
10. Dunlosky J, Rawson KA, Marsh EJ, Nathan MJ, Willingham DT. Improving students' learning with effective learning techniques: promising directions from cognitive and educational psychology. *Psychol Sci Public Interest*. 2013;14:4–58. doi:10.1177/1529100612453266. PMID:26173288.
11. Paddags B, Hershovich D, Savage V. Automated sentence generation for a spaced repetition software. In: *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*; 2024. p. 351–64.
12. Tural B, Örppek Z, Destan Z. Retrieval-augmented generation (RAG) and LLM integration. In: *Proceedings of the 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*; 2024. p. 1–5. doi:10.1109/ISAS64331.2024.10845308.
13. Team G, Georgiev P, Lei VI, Burnell R, Bai L, Gulati A, et al. Unlocking multimodal understanding across millions of tokens of context [Preprint]. 2024 Mar 8. arXiv:2403.05530.
14. Chen L, Zaharia M, Chen ZJ. FrugalGPT: how to use large language models while reducing cost and improving performance [Preprint]. 2023 May 9. arXiv:2305.05176.
15. Ong I, Almahairi A, Wu V, Chiang WL, Wu T, Gonzalez JE, et al. RouteLLM: learning to route LLMs with preference data [Preprint]. 2024. arXiv:2406.18665.
16. Huang J, Zhong Y, Chen X. Adaptive and personalized learning in STEM education using high-performance computing and artificial intelligence. *J Supercomput*. 2025;81:981. doi:10.1007/s11227-025-07481-7.
17. Nguyen A, Ngo HN, Hong Y, Dang B, Nguyen BT. Ethical principles for artificial intelligence in education. *Educ Inf Technol*. 2023;28:4221–41. doi:10.1007/s10639-022-11316-w. PMID:36254344.
18. Thabet B, Zanichelli N, Zanichelli F. Transformer-based recommender enabling automatic flashcards generation. In: *Book of abstracts: 4th International Conference on Higher Education Learning Methodologies and Technologies Online (HELMeTO 2022)*; 2022 Sep 21-23; Palermo, Italy. p. 222-5.
19. Henry MM, Heryanto NA, Isnain M, Nugroho KS, Pardamean B. Automatic multiple choice question generation: a systematic literature review. In: *Proceedings of the 9th International*

- Conference on Information Technology and Digital Applications (ICITDA); 2024. p. 1–7. doi:10.1109/ICITDA64560.2024.10810102.
20. Pawar P, Dube R, Joshi A, Gulhane Z, Patil R. Automated generation and evaluation of multiple-choice quizzes using LangChain and Gemini LLM. In: Proceedings of the International Conference on Electrical, Electronics and Computing Technologies (ICEECT). Vol. 1; 2024. p. 1–7. doi:10.1109/ICEECT61758.2024.10739326.