

AdaptIQ : NCERT-Aligned AI Learning Platform for Indian Students

Adithya M¹, Mithun Chavan A², Naga Tejas P³, Raja Naga Tejas⁴

^{1,2,3,4}Department of Computer Science and Engineering,

School of Computer Science & Engineering, Faculty of Engineering & Technology,
JAIN (Deemed-to-be University), Bangalore, Karnataka, India

¹22btrcn012@jainuniversity.ac.in, ²22btrcn171@jainuniversity.ac.in,

³22btrcn179@jainuniversity.ac.in, ⁴22btrcn223@jainuniversity.ac.in

Abstract— India has over 250 million school-going students assessed on the National Council of Educational Research and Training (NCERT) curriculum, yet no mainstream EdTech platform personalises learning at the level of individual chapters, exercises, or page references within actual textbooks. AdaptIQ V2 addresses this gap by integrating three scientifically validated techniques — the 2-Parameter Logistic Item Response Theory (IRT 2PL) model for real-time ability tracking, the SM-2 Spaced Repetition algorithm for long-term retention scheduling, and Retrieval-Augmented Generation (RAG) with pgvector for textbook-grounded AI tutoring — into a single NCERT-native platform for Classes 6–12. A five-step onboarding wizard captures twelve learner attributes; an adaptive quiz engine selects questions matched to each student's current ability score (θ), targeting a 65–70% success probability; and a daily study plan generator creates personalised schedules based on weak topic history and available study time. Additional features include a PDF viewer with an inline AI tutor that cites specific NCERT pages, a bilingual English/Hindi interface using official NCERT mathematical terminology, and a GitHub-style streak calendar for gamified engagement. Built on a React 18 + TypeScript frontend, Node.js/Express backend, and Supabase PostgreSQL with pgvector, AdaptIQ V2 delivers the first platform to combine NCERT alignment, IRT-based adaptive difficulty, SM-2 spaced repetition, and a textbook-aware AI tutor in a single production-grade system.

Keywords: adaptive learning, IRT 2PL, NCERT curriculum, spaced repetition, retrieval-augmented generation, EdTech, personalised education.

I. Introduction

The rapid adoption of digital learning platforms has transformed education globally; however, most existing systems deliver uniform content regardless of individual ability or curriculum context. In India, where over 250 million school-going students follow the NCERT syllabus for Central Board of Secondary Education (CBSE) and Indian Certificate of Secondary Education (ICSE) examinations, this gap is especially critical. Platforms such as BYJU'S, Unacademy, and Khan Academy provide broad content coverage but do not map learning interactions to specific NCERT chapters, exercises, or page references, leaving students unable to identify which textbook section to revisit after an

incorrect answer [5], [6].

AdaptIQ V2 is an AI-driven, NCERT-aligned adaptive learning platform designed to address these limitations. The system constructs a dynamic *Learner Profile* for each student — a continuously evolving ability score (θ) computed via the IRT 2PL model — and uses this profile to select questions, generate daily study plans, and ground AI tutoring in actual textbook content. Every interaction is anchored to a specific NCERT chapter and topic, creating a study experience genuinely relevant to board examination requirements.

A. Motivation

Five core limitations motivate the design of AdaptIQ V2: (1) existing platforms lack curriculum-native question mapping; (2) no Indian EdTech tool dynamically tracks student ability at sub-question granularity; (3) spaced repetition is absent from NCERT-aligned platforms; (4) AI tutors cannot cite specific NCERT page references; and (5) personalized daily plans grounded in textbook content are unavailable to the average Indian student.

B. Contributions

This paper makes the following contributions:

- A full NCERT curriculum hierarchy of 45 chapters across Classes 8–10 Mathematics, with bilingual English/Hindi titles.
- A real-time IRT 2PL adaptive quiz engine targeting a psychometrically optimal 65–70% success probability.
- An SM-2 spaced repetition scheduler at question level with `nextReviewAt` timestamps.
- A RAG-based AI tutor grounded in cached NCERT page text via pgvector.
- A personalized *Today's Plan* generator integrating onboarding data and real-time performance history.

II. Literature Survey

A. Generic EdTech Platforms

Platforms such as BYJU'S, Khan Academy, and Unacademy provide wide content coverage through video lessons and static quizzes. However, they follow a one-size-fits-all approach and are not mapped to NCERT chapters, exercises, or page references [8]. A student who answers a question incorrectly has no mechanism to identify the corresponding textbook section for remediation.

B. Item Response Theory in Adaptive Systems

The 2-Parameter Logistic IRT model, established in psychometrics literature [1], computes the probability of a correct response as:

$$P(\text{correct} \mid \vartheta, a, b) = \frac{1}{1 + e^{-a(\vartheta - b)}} \quad (1)$$

where ϑ is learner ability, b is item difficulty, and a is item discrimination. Computer-adaptive tests such as the GRE and GMAT rely on IRT, but existing IRT-based systems lack alignment to the Indian curriculum or board examination patterns.

C. Spaced Repetition Systems

The SM-2 algorithm [2] schedules review intervals based on past performance to maximize long-term retention. Platforms such as Anki and Duolingo apply SM-2 effectively; however, these are not contextualised to the NCERT syllabus, and no existing tool integrates spaced repetition with curriculum-mapped question banks for Indian school students.

D. Retrieval-Augmented Generation for Tutoring

RAG systems [3] combine dense vector retrieval with large language models (LLMs) to ground generated responses in factual documents. General-purpose AI tutors such as ChatGPT and Khanmigo answer student questions but cannot reference specific NCERT examples by exercise number or cite the exact page where a concept appears.

E. Learning Analytics and Gamification

Learning analytics research [6], [14] demonstrates that dashboards and progress trackers help identify struggling students. Gamification studies [7] confirm that streak-based mechanics and visual feedback loops improve daily active usage. AdaptIQ V2 applies these findings through a GitHub-style heatmap calendar and streak statistics.

F. Gap Analysis

As summarized in Table I, no existing platform simultaneously provides NCERT-mapped questions, IRT-based adaptive difficulty, SM-2 spaced repetition, and RAG-powered textbook-referenced AI tutoring for Indian school students.

III. Methodology

AdaptIQ V2 follows a modular, layered architecture in which each component communicates through well-defined TypeScript interfaces.

A. System Architecture

The platform comprises four main layers:

1) *Frontend Layer* React 18 + TypeScript + Tailwind-CSS provides the student-facing dashboard, onboarding wizard, quiz interface, PDF viewer, and streak calendar. Vite is used as the build tool; Framer Motion handles animations; `react-pdf` renders NCERT textbook pages in-browser.

TABLE I. Comparative Analysis: Existing Systems vs. AdaptIQ

System	Strength	Limitation
BYJU'S / Unacademy	Wide content coverage	No NCERT mapping
Khan Academy	Mastery-based flow	Not NCERT aligned
IRT-based CAT	Adapts to ability	No Indian curriculum
Anki / SM-2	Proven retention	No NCERT context
ChatGPT / Khanmigo	Answers questions	No chapter/page ref.
AdaptIQ V2	All above + NCERT	First of its kind

2) *Backend Layer* A Node.js + Express 5 + TypeScript REST API implements the IRT scoring engine (`irt.ts`), SM-2 spaced repetition scheduler (`spaced-repetition.ts`), quiz session lifecycle, NCERT content routing, today's plan generator, and AI tutoring proxy.

3) *AI Layer* OpenRouter routes LLM requests to *claude-sonnet*. For daily plan generation, the AI receives the student's Learner Profile and available NCERT chapters; for tutoring, it receives cached page text from `NCERTPDFTextCache` and an NCERT-aware system prompt, then returns explanations citing specific chapter and page context.

4) *Database Layer* Supabase (managed PostgreSQL + pgvector) stores user records, Learner Profiles, NCERT curriculum hierarchy, question banks, quiz sessions, study plans, and daily activity logs. Prisma ORM manages schema migrations.

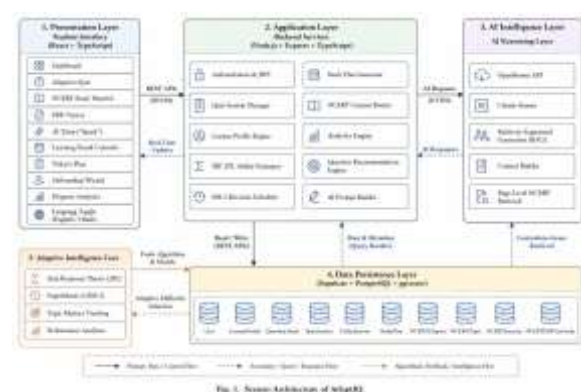


Fig. 1. System Architecture of AdaptIQ V2. The five-layer architecture separates UI, business logic, AI reasoning, and persistent storage, communicating via typed REST APIs.

B. Student Onboarding

A five-step animated wizard collects 12 data points per student: class and board, study goals (e.g., score 90%+ in boards), daily study time (15/30/60/120 min), learning style (visual, examples, reading, mixed), weak

topics, and target score. These are persisted in the `LearnerProfile` table and seed all downstream personalisation.

C. Adaptive Quiz Engine

Each quiz session calls `/api/questions/next`, which selects the question whose difficulty best matches the student's current ability score ϑ , targeting $P(\text{correct}) \approx 0.65\text{--}0.70$ — the psychometrically optimal challenge level for learning [1].

D. Algorithms

1) *IRT 2PL Ability Update* After each response, ability is updated via a Newton–Raphson-inspired gradient step:

$$\vartheta' = \vartheta + \alpha \cdot a \cdot r - P(\vartheta) \quad (2)$$

where α is the learning rate, $r \in \{0, 1\}$ is the binary response, and $P(\vartheta)$ is the IRT prediction from Eq. (1). Difficulty parameters are mapped from categorical labels: easy ($b = -1.0, a = 0.8$), medium ($b = 0.0, a = 1.2$), hard ($b = 1.5, a = 1.5$). The ability score is clamped to $[-3, +3]$.

2) *SM-2 Spaced Repetition* The inter-repetition interval $l(n)$ follows the SM-2 schedule [2]:

$$\begin{aligned} l(1) &= 1 \text{ day}, \quad l(2) = 6 \text{ days} \\ l(n) &= l(n-1) \times EF, \quad n > 2 \end{aligned} \quad (3)$$

where the Ease Factor is updated as:

$$EF' = EF + 0.1 - (5 - q)(0.08 + (5 - q) \times 0.02) \quad (4)$$

and quality score $q \in \{0, \dots, 5\}$ reflects recall quality. If $q < 3$, the interval resets to 1 day, re-surfacing missed content at optimal intervals.

E. NCERT Curriculum Integration

Four new database tables — `NCERTChapter`, `NCERTTopic`, `NCERTExercise`, and `NCERTPDFTextCache` — mirror the NCERT textbook structure. All 45 chapters across Classes 8, 9, and 10 Mathematics are seeded with bilingual titles. A bulk mapping script (`mapQuestionsToNCERT.ts`) assigns NCERT metadata to questions based on topic slugs; an admin UI supports manual review. During quizzes, mapped questions display a contextual NCERT chip showing class, chapter, and page reference.

F. Today's Plan Generator

The `/api/plan/today` endpoint constructs a prioritised list of 2–4 study items (weak topic review, NCERT chapter reading, adaptive quiz) tailored to the student's available daily time and learning style, then persists the plan as a JSONB object in `StudyPlan`. Students mark items complete from the dashboard; progress survives page refreshes. Plans are regenerated daily and never duplicated.

G. PDF Viewer with AI Tutor

NCERT PDFs are served from the backend and rendered page-by-page via `react-pdf`. The AI chat

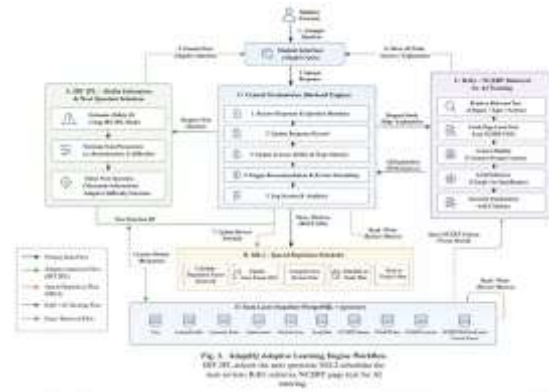


Fig. 2. AdaptIQ Adaptive Learning Engine Workflow.

IRT 2PL selects the next question; SM-2 schedules the next review; RAG retrieves NCERT page text for AI tutoring.

sidebar calls `/api/ai/explain`, which retrieves cached page text from `NCERTPDFTextCache`, constructs an NCERT-aware system prompt (chapter title, class, subject, page text), and streams a contextual explanation referencing specific NCERT examples. Conversation history is maintained within the session for multi-turn tutoring.

IV. Results

AdaptIQ was built, integrated, and validated end-to-end. All major components operate as intended and communicate through shared TypeScript type contracts.

A. Onboarding and Learner Profile

The five-step onboarding wizard successfully captures all 12 data points. Upon completion, `User` and `LearnerProfile` records in Supabase are fully populated. The flow includes a skip option and auto-redirects new users after login.

B. NCERT Curriculum Browser

The study material page lists all 45 NCERT chapters across Classes 8, 9, and 10 Mathematics (15 per class), each displaying bilingual titles, estimated study hours, and chapter number. The language switcher allows global toggle between English and Hindi content. Chapter cards navigate directly to the PDF viewer.

C. Adaptive Quiz Engine

The IRT 2PL engine dynamically adapts difficulty across a 10-question session: a student answering correctly receives progressively harder questions; incorrect answers trigger easier follow-ups. Post-session summaries display total accuracy, average time per question, best streak, and topic-level breakdown. The SM-2 scheduler correctly sets `nextReviewAt` timestamps and re-surfaces missed questions at appropriate intervals.

D. Today's Plan and Streak Calendar

The dashboard Today's Plan card displays 2–4 personalised study items with estimated time, type (chap-

ter/quiz/review), and a live progress bar. The GitHub-style 6-month heatmap displays daily study intensity with five color levels (gray = no activity; dark green = 20+ questions). Current streak, longest streak, and total study days are surfaced with adaptive motivational messages.

E. PDF Viewer and AI Tutor

Students open any NCERT chapter and read the textbook in-app. The AI tutor responds with references to specific NCERT examples on the current page — a capability unavailable in any competing platform.

F. End-to-End Integration Summary

Table II summarises the completion status of all V2 deliverables.

TABLE II. V2 Feature Delivery Status

Feature	Backend	Frontend
5-step onboarding wizard	✓	✓
NCERT hierarchy (45 chapters)	✓	✓
IRT 2PL adaptive quiz engine	✓	✓
SM-2 spaced repetition	✓	Transparent
Today's Plan generator	✓	✓
PDF viewer + AI tutor	✓	✓
Learning streak calendar	✓	✓
Question-NCERT mapping	✓	✓
Bilingual English/Hindi support	✓	✓

V. Discussion

The results confirm that combining IRT 2PL, SM-2 spaced repetition, and RAG-powered tutoring within a NCERT-native architecture yields a qualitatively different learning experience compared to existing Indian EdTech platforms. The IRT ability score provides sub-question granularity unavailable in mastery-based or video-first systems. The SM-2 scheduler replicates a core function of a skilled human tutor: reviewing content at the precise moment retention begins to fade [2]. The RAG AI tutor overcomes the principal limitation of general-purpose chatbots by citing specific textbook examples rather than generating unconstrained responses [3].

The bilingual interface — using NCERT's own Hindi mathematical terminology — reduces the cognitive load for Hindi-medium students, a demographic largely underserved by English-first EdTech products. The gamified streak calendar applies evidence-backed engagement mechanics [7] within a curriculum-aligned context, encouraging consistent daily practice rather than passive video consumption.

Limitations of the current implementation include the NCERT hierarchy being limited to Classes 8–10 Mathematics; full semantic pgvector RAG (embedding-

level retrieval across all textbook pages) is planned but not yet deployed. The platform currently supports individual student use; parent and teacher dashboards with progress digests remain future work.

VI. Conclusion

AdaptIQ V2 demonstrates that a NCERT-native, AI-driven adaptive learning platform is technically feasible using modern open-source tooling. By anchoring every interaction — question selection, daily plan, AI explanation — to a specific chapter and page in the actual NCERT textbook, AdaptIQ V2 creates a study experience genuinely relevant to a student's board examination requirements.

The integration of IRT 2PL ability tracking, SM-2 spaced repetition, and RAG-powered AI tutoring into a single bilingual platform represents the first of its kind for the Indian school education segment. The system is deployed using Supabase Cloud, OpenRouter, and Vercel/Railway — a fully managed, horizontally scalable stack accessible to any student with a web browser.

Future work includes expanding the curriculum to Classes 6–7 and Classes 11–12 (JEE/NEET foundation), enabling full pgvector semantic retrieval across all NCERT pages, building parent/teacher dashboards, and implementing a board exam simulation mode with CBSE marking scheme awareness.

Acknowledgement

The authors express sincere gratitude to Mr. Ravi Lanke Kumar, Assistant Professor, Department of Computer Science and Engineering, JAIN (Deemed-to-be University), for his invaluable guidance throughout the project. The authors also thank Dr. Mahesh T.R., Program Head, and Dr. Kamlesh Tiwari, Head of Department, for their continuous academic support.

References

- [1] F. Baker and S. Kim, *Item Response Theory: Parameter Estimation Techniques*, 2nd ed. Marcel Dekker, New York, 2004.
- [2] P. Wozniak, "SuperMemo 2: Optimal Intervals for Spaced Repetition," *Acta Neurobiologiae Experimentalis*, 1994.
- [3] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Proc. NeurIPS 2020*, arXiv:2005.11401.
- [4] NCERT, *Mathematics Textbooks Classes 6–10*. National Council of Educational Research and Training, New Delhi, 2024.
- [5] P. Brusilovsky and E. Millán, "User Models for Adaptive Hypermedia and Adaptive Educational Systems," in *The Adaptive Web*, ser. Lecture Notes in Computer Science, vol. 4321. Springer, 2007, pp. 3–53.
- [6] G. Siemens and P. Long, "Penetrating the Fog: Analytics in Learning and Education," *EDUCAUSE Review*, vol. 46, no. 5, pp. 30–40, 2011.
- [7] J. Hamari, J. Koivisto, and H. Sarsa, "Does Gamification Work? A Literature Review of Empirical Studies on Gamification," in *Proc. 47th Hawaii Int. Conf. System Sciences (HICSS)*, 2014.

- [8] K. VanLehn, "The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems," *Educational Psychologist*, vol. 46, no. 4, pp. 197–221, 2011.
- [9] React Developers, "React Documentation," Meta Platforms Inc. [Online]. Available: <https://react.dev>. [Accessed: 2025].
- [10] Supabase, "Supabase Documentation." [Online]. Available: <https://supabase.com/docs>. [Accessed: 2025].
- [11] OpenRouter, "OpenRouter API Documentation." [Online]. Available: <https://openrouter.ai/docs>. [Accessed: 2025].
- [12] pgvector, "Open-Source Vector Similarity Search for Postgres." [Online]. Available: <https://github.com/pgvector/pgvector>. [Accessed: 2025].
- [13] Prisma, "Prisma ORM Documentation." [Online]. Available: <https://www.prisma.io/docs>. [Accessed: 2025].
- [14] R. Ferguson, "Learning Analytics: Drivers, Developments and Challenges," *Int. J. Technology Enhanced Learning*, vol. 4, no. 5/6, pp. 304–317, 2012.