VERITAS

Building Veritas, a learning app that checks what you remember

Veritas reads any book or document to you in your own voice, quizzes you at the end of each chapter, and brings back what you got wrong until it sticks. I started it and designed and built the platform solo over 20 months, including the research, the design, the AI decisions and the launch plan. It launches in Q1 2027 and isn't publicly released yet, so this case study covers the thinking and decisions, plus early results from a closed beta and an offline evaluation.

BUILD
Solo, over 20 months

STATUS
Launching Q1 2027

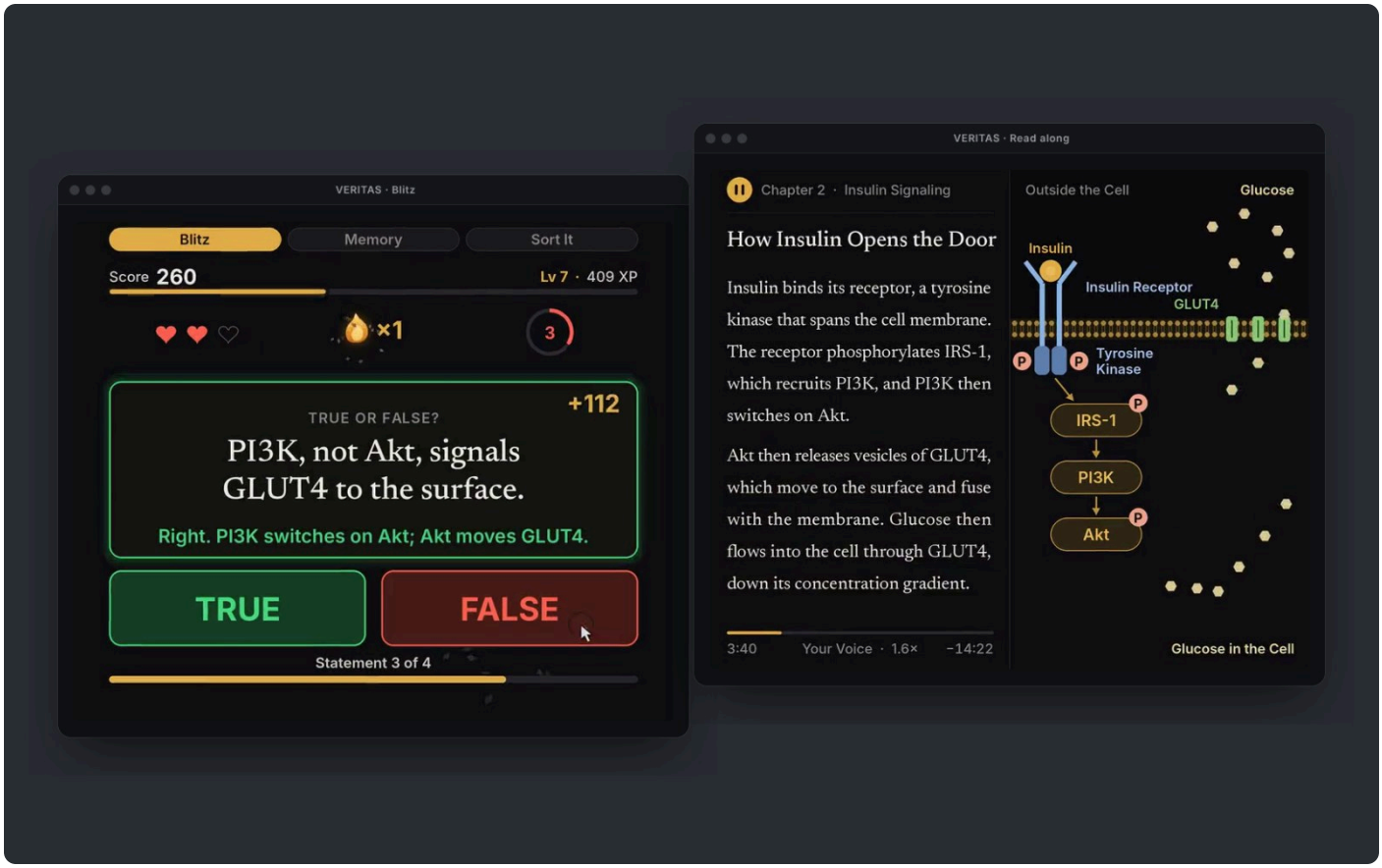
Veritas website **profio1997.github.io/Veritas_Website/** · Veritas on LinkedIn **www.linkedin.com/company/veritas-elearning/**

2,500+
on the waitlist, with no paid marketing

85%
of 60 closed-beta students returned for daily review in week 2

97% / 99%
source-grounded answers and answer-key accuracy on 400+ real chapters

<$0.30
AI cost per book, with chapter-ready audio in 45 seconds



Illustrative.

Overview

Veritas started because we forget much of what we read. When I looked into why, the research was clear. Most of what people do to study (rereading, highlighting, summarizing) barely helps. The things that do help are testing yourself, spacing your practice and explaining things in your own words, and those are hard to keep up alone.

So I'm building one product that handles the dull parts. You bring in a book, lecture notes or a long report. Veritas reads it to you in your own voice, checks what you took in at the end of each chapter, and brings back what you missed a little before you'd have forgotten it. I do the product work myself, which covers the research, the design of every step, the decisions about where AI belongs and the launch plan.

With the same passage and the same study time, people who read once and recalled it three times remembered 61% a week later. People who reread it four times remembered 40% (Roediger and Karpicke, 2006).

Problem and research

Three findings shaped everything that followed.

First, reading itself is in decline. A 2025 study in iScience (Bone et al.) found that the share of Americans who read for pleasure on a given day fell from 28% in 2004 to 16% in 2023. A 2026 report from Cornell's Committee on the Future of the American University said students raised on digital technology are quick to find information but struggle with sustained reading and complex arguments.

Second, much of what people do with what they read doesn't work. In a 2013 review of ten common study techniques, Dunlosky and colleagues rated rereading, highlighting and summarizing as low utility. Practice testing and spaced practice rated high.

Third, forgetting is predictable. Ebbinghaus described the curve in 1885 and Murre and Dros replicated it in 2015. Material you never revisit fades quickly, and each well-timed review flattens the curve.

Put together, the evidence pointed me away from faster reading and toward practice. People need to rehearse remembering the things they have to know, at the right moments.

> *"The tools stop at the first step. Reading apps stop at listening, and study apps live somewhere else and never see what you read."*

Who it's for

I started with three groups, since they read for different reasons and put up with different amounts of effort. I'm building for the first one first.

1. **Students with an exam ahead**
   They have a fixed deadline, a lot of text and not enough time. They want to listen on the bus, take a check when they sit down, and review weak points until the exam. Textbooks, lecture notes and papers.

2. **Professionals learning on the job**
   They need to understand a report or technical PDF quickly and still know it next quarter. They want the fastest speed they can follow, a place for their own notes, and review that fits around work.

3. **Lifelong learners**
   They own the books they've always meant to finish, like Meditations, The Art of War or Walden. They want to finish able to explain the book to someone else.

What already exists

I looked at four kinds of tools: read-aloud and audiobook apps, note-taking and study tools that work on your own files, flashcard apps, and quiz and game apps. Each is good at its own step. None of them knows what you missed in the step before.

TO MASTER WHAT YOU READ	READ-ALOUD APPS	NOTE APPS	FLASHCARD AND QUIZ APPS	VERITAS
Reads any document aloud, every word highlighted	✓Yes	–No	–No	✓Yes
Reads in your own voice	Some, paid	–No	–No	✓Yes
Your own notes, chapter by chapter	Partly	Partly	–No	✓Yes
Questions made from what you read	Some	✓Yes	Partly	✓Yes
A check every chapter that finds weak points	–No	Partly	–No	✓Yes
What you miss comes back in practice and review	–No	–No	–No	✓Yes
Spaced review across everything you read	–No	–No	Cards you have	✓Yes
A plan with a finish date for each book	–No	–No	–No	✓Yes

The last three rows matter most. A flashcard app can space review well, but only for cards you made. A read-aloud app can read anything and has no idea whether you understood it. I think there's room in connecting the two, so what you get wrong in a chapter check shapes tomorrow's review.

Solution

Principles

Before I drew a screen I wrote down the rules I'd use to say yes or no to features. They come from the research.

A feature has to help people remember

Every format in the product is a form of practice testing, generative learning (like drawing or teaching back, from Fiorella and Mayer's 2016 review) or spacing. If an idea is popular but doesn't strengthen recall, it waits.

Start from the reader's own material

A curated library would be easier to build and quality-check. But people study what they have to study, so the product takes PDFs, EPUBs, Word files, web pages, photos and scans, and finds the chapters itself.

Your own voice, by default

Reading six short passages aloud, about two minutes, is enough to build a voice that reads any book back to you. It's the first thing people react to, and it gives them a reason to bring their whole library in. Preset voices are there for anyone who'd rather not record.

Don't write the notes

Summaries are the feature people ask for most, and the research rates summarizing as low value for remembering. So each chapter starts with an empty note, and Veritas never writes or inserts notes for you. This is the call I'm least sure everyone will accept.

Fit the product to the reader

A short speed test finds the fastest pace at which you still understand. A short mode test compares reading, read-along and audio-only on real passages and sets your default from your own scores. The defaults come from how you learn instead of a settings screen.

Close the loop

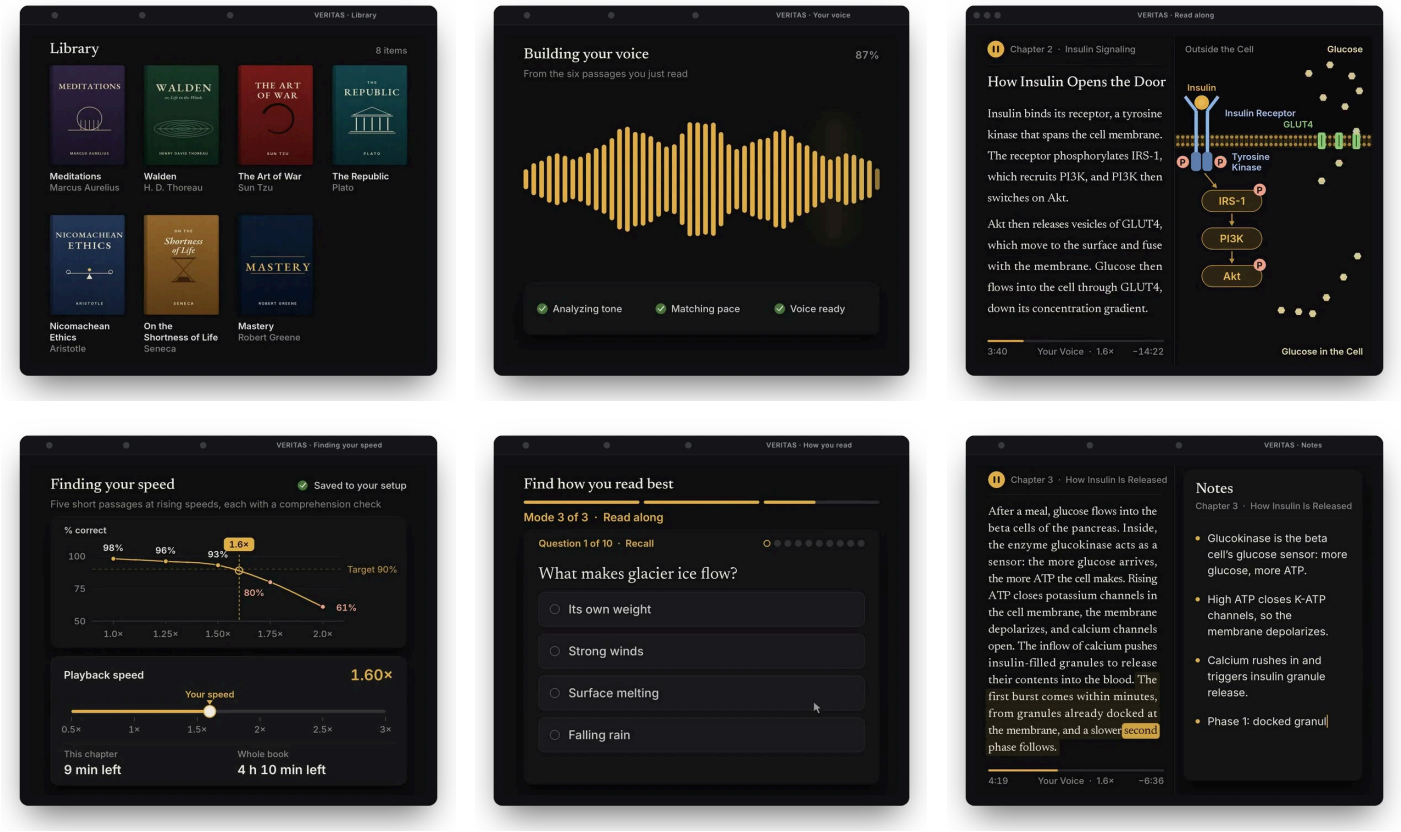
What you miss in a check should come back in practice and in review. That connection is the product, so I'd cut a feature before I'd break it.

The product, step by step

The research points to three things for long-term memory: encode the material well, practice retrieving it, and space that practice out. Veritas runs all three on anything you read. Below is how each step works and what I decided along the way.

Step 1. Encode

It starts in the library, which takes PDFs, EPUBs, Word files, web links, photos and scans, or pasted text, organized on shelves and in collections. Chapters are found for you. Then the reader hears the text with every word highlighted and a picture of the idea building beside it, following Mayer's multimedia principle that people learn better from words and pictures than from words alone.

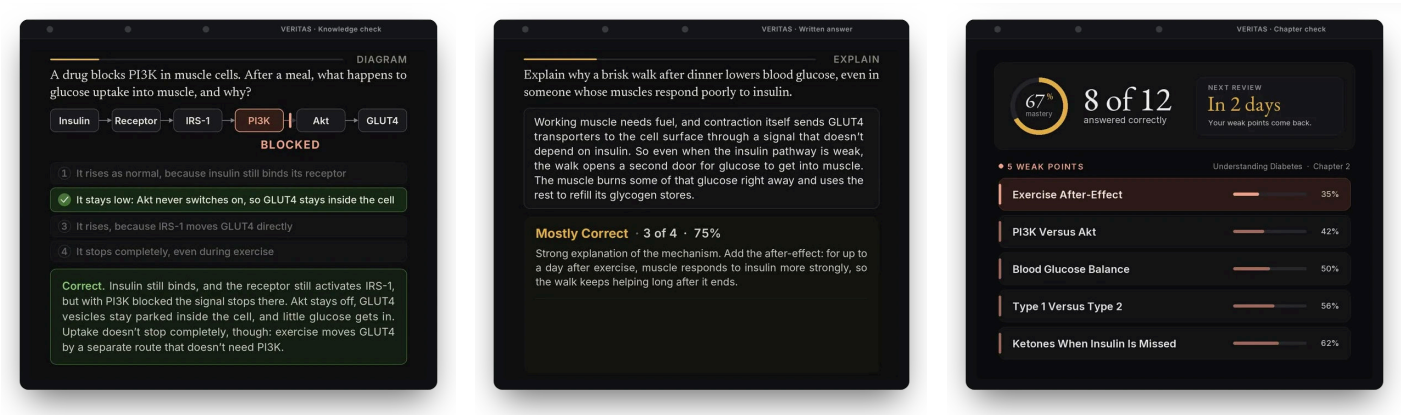


Illustrative.

The speed test is a small decision with a big effect. Most apps let you pick a playback speed and leave it, but people overestimate how fast they can listen and understand. So the test reads short passages at increasing speeds, each followed by a comprehension question, and plots the result. It shows where understanding starts to fall, then how long the current chapter and the whole book will take at that pace. Chapter recaps are also read aloud in the reader's voice, with sections they can jump between.

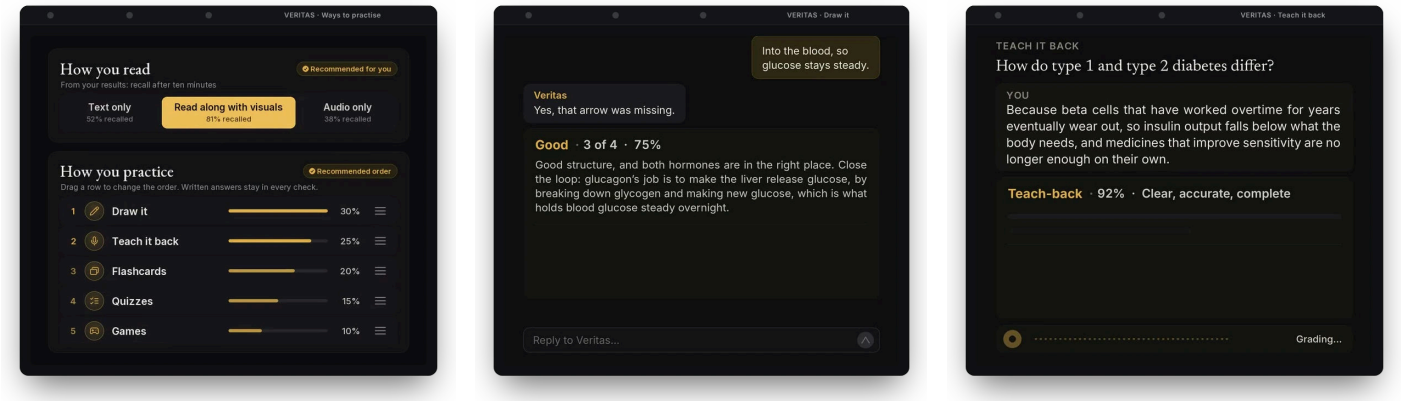
Step 2. Retrieve

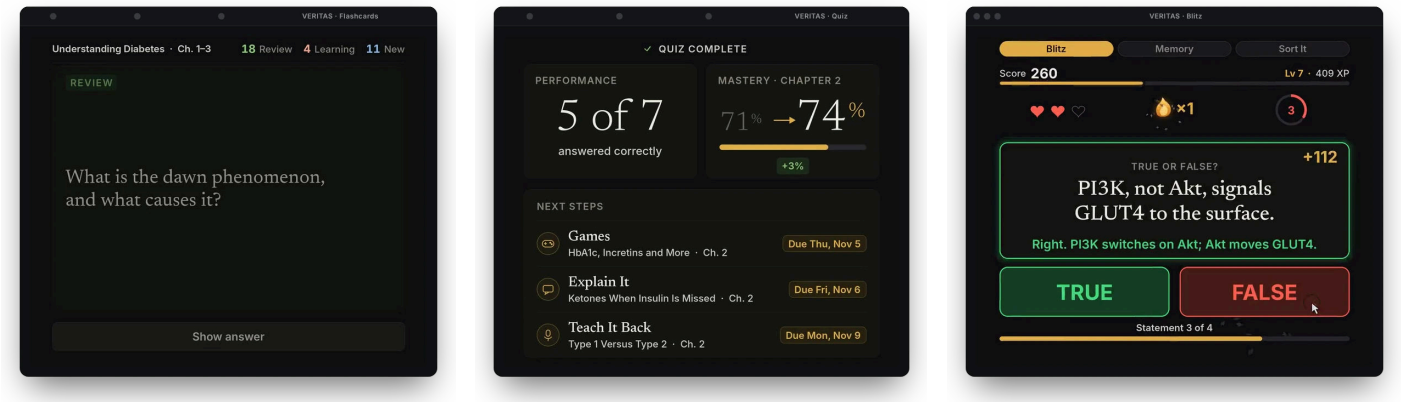
Every chapter ends with a knowledge check of about 12 mixed questions that read like a real exam, diagrams included. It takes about 12 minutes. After each answer the product explains why a wrong option is wrong, then why the right one is right. Written questions ask the reader to explain something in their own words, and Veritas says what they covered and what they missed.



Illustrative.

The weak points then drive practice. The reader can work on them in six main ways: written answers, multiple choice, drawing from memory, teaching the topic back out loud, flashcards and games. Veritas recommends one based on the reader's own results, and the reader can reorder the list. I included drawing and teaching back because the evidence for them is strong. Explaining an idea, to yourself or to a friend, exposes gaps that recognizing the right answer hides.

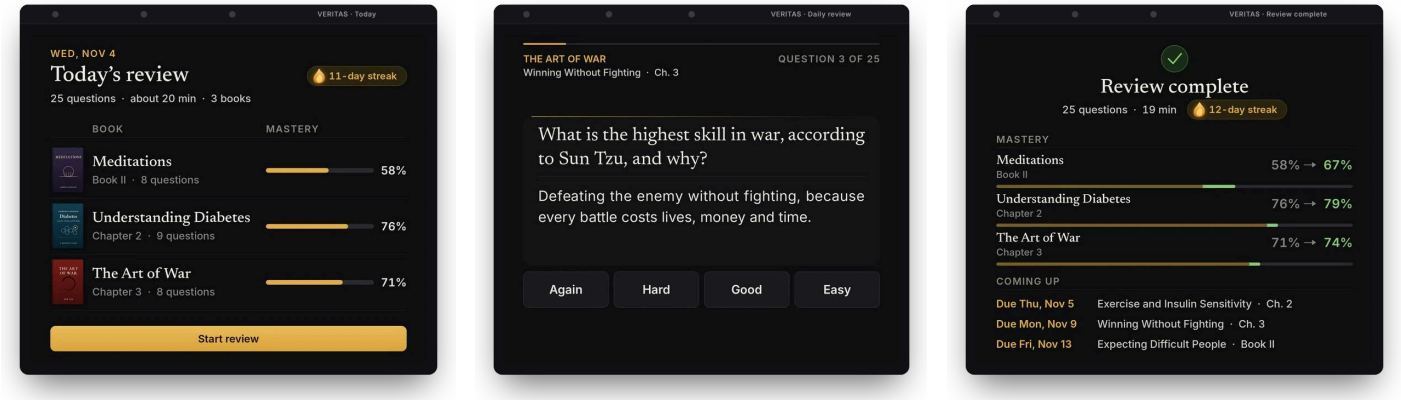




Illustrative.

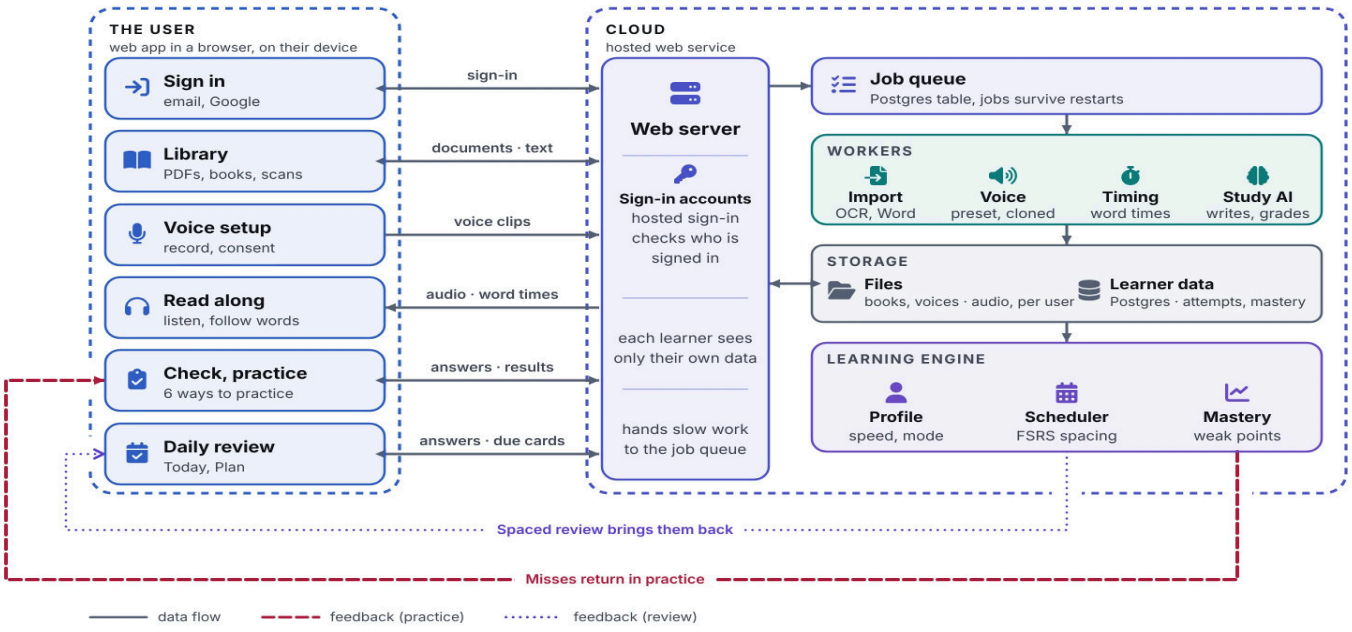
Step 3. Space it out

The third step is meant to turn the product into a habit. Instead of a review queue for every book there's one daily review across all of them, in a calm full-screen mode. What you miss comes back sooner and what you know comes back less often. Before you start, the Today screen says what's due, how many questions it is and about how long it will take. I figure a review you can't size up is one you put off.



Illustrative.

How the platform works



Illustrative.

Learners only ever use the web app in their browser: the library, voice setup, read-along, check and practice, and the daily review. Sign-in is hosted, so Veritas never stores a password. The browser also records the voice clips, and it asks for consent before it does.

Everything else runs in the cloud. The web server checks who is signed in, shows each learner only their own data, and hands slow work to a job queue. The queue is a table in Postgres, so jobs survive a restart. Four workers take jobs from it: import (OCR and Word conversion), voice (preset and cloned voices), timing (word times for read-along) and the study model that writes and grades questions. Files are kept per learner, and learner data in Postgres has an owner on every row.

The learning loop runs on top. A check of about 12 mixed questions closes each chapter, and written answers are graded against the source passage. What you miss feeds six ways to practice, and a spaced scheduler (FSRS) brings it back in one daily review across everything you read. The daily review ran in the closed beta. The fuller learning engine (profile, scheduler and mastery tracking) is still being built out.

Where the AI is, and how I'm checking it

Most of the product is ordinary software. The AI does four jobs, and in each one I've decided in advance what it is allowed to do.

FEATURE	WHAT THE MODEL DOES	HOW I'M CHECKING IT
Your-voice reading	Builds a voice from the reader's recording	Only the reader's own voice, from a recording they make on purpose. Preset voices are an alternative.
Questions and flashcards	Writes questions from the reader's text, including diagram questions	A question must be answerable from that text, and every answer is explained.
Feedback on answers	Says what a written, drawn or spoken answer covered and missed	Compared against the source passage, not against general knowledge.
Review schedule	Decides when weak points come back	The reader sees what's due and how long it will take before starting.

The main risk in a product like this is a confident wrong answer, such as a question with the wrong key or feedback that calls a correct explanation incorrect. A learner might not notice and would end up studying the mistake. So evaluation is a launch requirement. Before any question type or feedback type ships, it has to pass a review set built from real chapters across subjects. The set checks whether the answer is in the text, whether the key is right and whether the explanation helps. The set now covers 400+ real chapters across 12 subjects, and the product has cleared all of its launch gates (see Results). After launch I'll track how often learners flag a question or dispute feedback, and fix patterns by question type.

WHAT I'M MEASURING	BEFORE LAUNCH	AFTER LAUNCH
Question quality	Answer is in the source; key is correct; difficulty spread on a review set of real chapters	Flags per 1,000 questions
Feedback accuracy	Agreement with a rubric on written, drawn and spoken answers	Disputed feedback rate
Learning	A pilot against a rereading baseline	Mastery gain per book; return to the daily review in week four

Two more choices. The library and the voice belong to the reader and are used only to serve that reader. And I won't make claims about learning outcomes until I've measured them.

Results

Veritas isn't publicly released, so these are early results from a waitlist, a closed beta and an offline evaluation, not results from a public launch. I built the platform solo over 20 months: OCR import, voice cloning, read-along timing and an LLM study model running on a durable Postgres queue.

NUMBER	WHAT IT IS	HOW TO READ IT
2,500+ on the waitlist	People who signed up for early access, with no paid marketing	Interest, not usage. It is where the first audience will come from.
3 university study-skills centers	Signed on as pilot partners	Partners for the first pilot. There are no pilot results yet.
60-student closed beta	Students who used the product before launch	A small group, so I read it as an early signal and not a measurement of learning.
85% returned for daily review in week 2	Share of the beta students who came back to the daily review in the second week	Coming back to the daily review is the habit the product depends on.
4.4 out of 5 for voice naturalness	How the beta students rated their cloned voice, close to how they rated their own recordings	Ratings from the closed beta.
97% source-grounded, 99% answer-key accuracy	An offline evaluation on 400+ real chapters across 12 subjects	The product cleared all of its launch gates. It checks that answers come from the chapter and that the key is right. It doesn't measure whether learners remember more.
Under $0.30 per book, 45 seconds to chapter-ready audio	AI cost per book and the time to get a chapter's audio ready	Cost and speed of the current build.

Launch and open questions

Launching

I'm running the launch in public stages. A pre-release website explains the product, with a 2 minute 32 second film that walks through the whole loop in seven chapters, from "why we forget" to "one platform and why now," and a waitlist for early access. The first audience is students with an exam ahead, because the need is urgent and the habit is easiest to form when there's a date. Pricing will be announced before launch, and people on the waitlist hear first. Public launch is planned for Q1 2027, after the first pilot.

MILESTONE	STATUS
Pre-release site, 2:32 film and waitlist	Live, 2,500+ on the waitlist
Designed screens for all three steps	**Done**
Pre-release build	Used in a closed beta with 60 students
Offline evaluation set and launch gates	400+ chapters across 12 subjects, all launch gates cleared
First pilot with real learners against a rereading baseline	Planned before launch, with 3 university study-skills centers signed on
Public launch	**Q1 2027**

What I don't know yet

- Whether people learn more with Veritas than by rereading. I'll measure this against a rereading baseline in the pilot, and until then I'm only claiming what the published research supports.
- Whether readers will accept a product that won't write their notes. If it costs us sign-ups I'll learn from that, but I don't want to drop it without evidence.

- How much effort people will put into a daily review. 85% of the closed-beta students came back in week 2, which is an encouraging sign, but week 2 is early. If the Today screen doesn't keep people coming back, the product doesn't work, however good the questions are.
- How well the voice works across accents and recording conditions.

What I learned

- Starting from published research gave me one rule for the roadmap. If a feature doesn't help people remember, it waits. That has made saying no easier.
- The demo explains the product better than any description does. Once people see a check, then targeted practice, then tomorrow's review, it makes sense.
- Because I built it alone over 20 months, I learned to be careful with the hard parts (evaluation, the review schedule) and keep the rest simple.