

unvibe

PRIVATE BETA

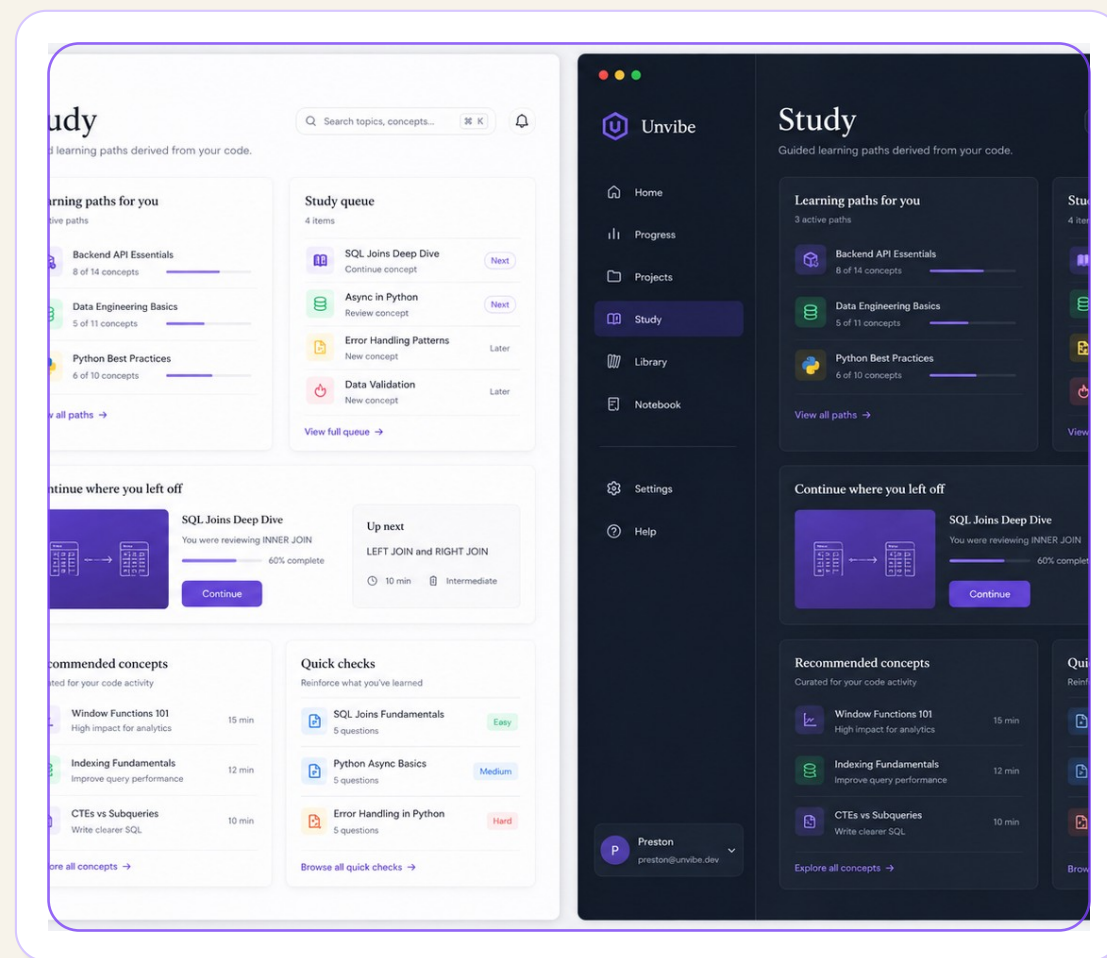


Understand the code AI writes.

Make the code yours.

Unvibe is a desktop learning layer that explains selected code, project changes, and repositories beside the tools where people already build.

Join the private beta



AI writes the code. Unvibe helps you understand it.

AI-assisted coding is becoming the default workflow.

The next gap is not code generation - it is code ownership: can users explain, maintain, and safely change what agents produce?

80%

of new developers on GitHub use Copilot within their first week, per GitHub Octoverse 2025.

>75%

developers in IDC's 2024 survey reported productivity improvement from GenAI coding tools.

Generate

Agent writes feature

Agent fixes error

Agent refactors file

Agent adds dependency



Own

What changed?

Why this pattern?

What can break?

How do I rebuild it?

Sources: GitHub Octoverse 2025; IDC Generative AI in the SDLC paper, based on 2024 GenAI Developer Trends Survey.

Your agent finished the feature. Could you explain it tomorrow?

Using AI is not the problem. Losing the thread of your own project is.

Before Unvibe

- Large AI-generated diff
- Unfamiliar abstractions
- Hidden failure points
- "It works, I think."

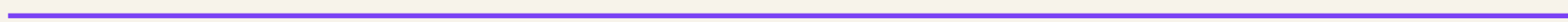
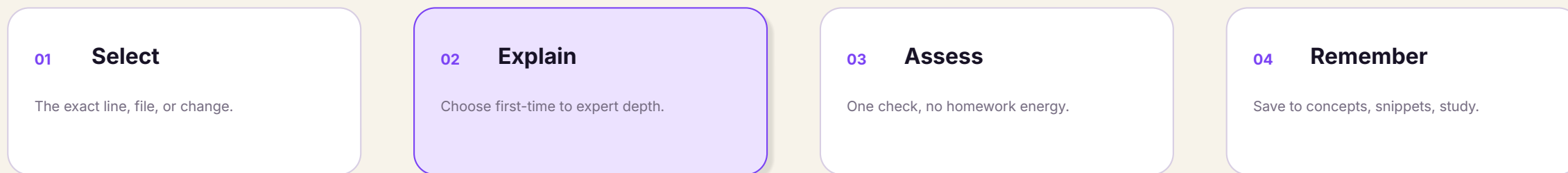


After Unvibe

- What changed
- Why it exists
- Files + concepts involved
- One check to lock it in

A quiet loop from selected code to retained knowledge.

Unvibe lives as a small overlay, then saves the important parts into a learning profile.



The flywheel: each explanation makes the next explanation more personal.

Native-feeling help that stays beside the work.

The interface is intentionally small: select code, pick depth, read, save, continue coding.

Small overlay

Does not replace the editor

Depth picker

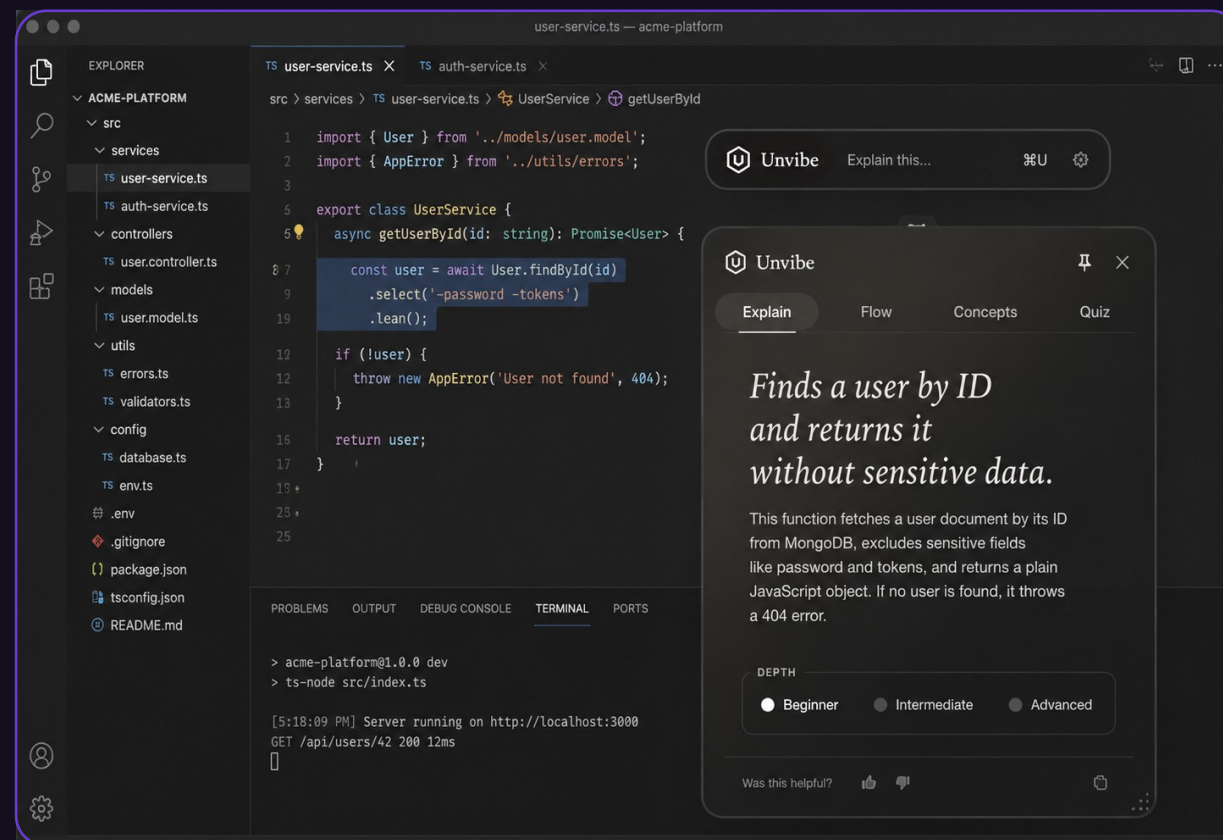
First-time → Expert

Actionable answer

Why, how, risks, example

Memory

Save concepts and snippets

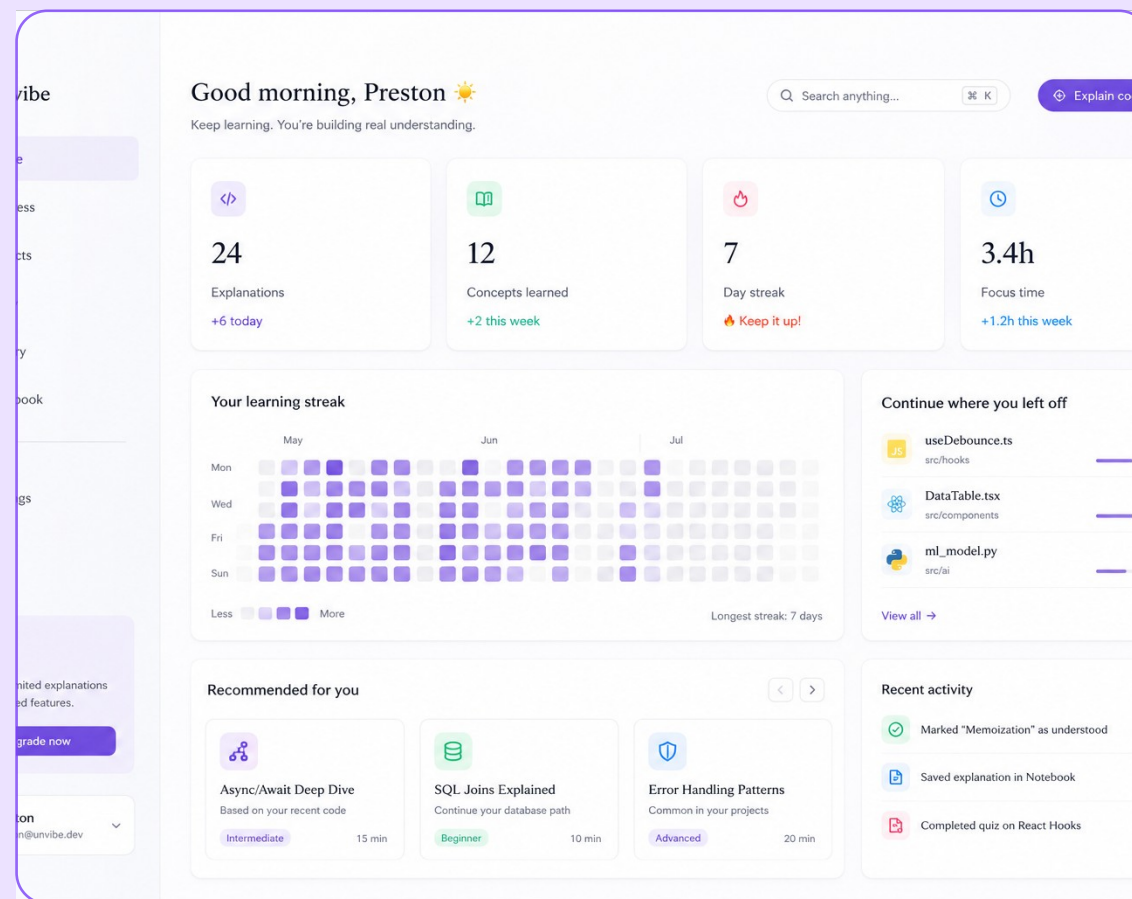


A line rarely makes sense by itself.

Unvibe moves outward from expression to architecture, then turns missing context into something learnable.



No generic tutorial can know the decisions inside your repository.



The moat is not one explanation. It is evidence over time.

Unvibe builds a learning profile from reviewed code, saved concepts, checks, repeated questions, and projects.

React state

Familiar

SQL joins

Needs review

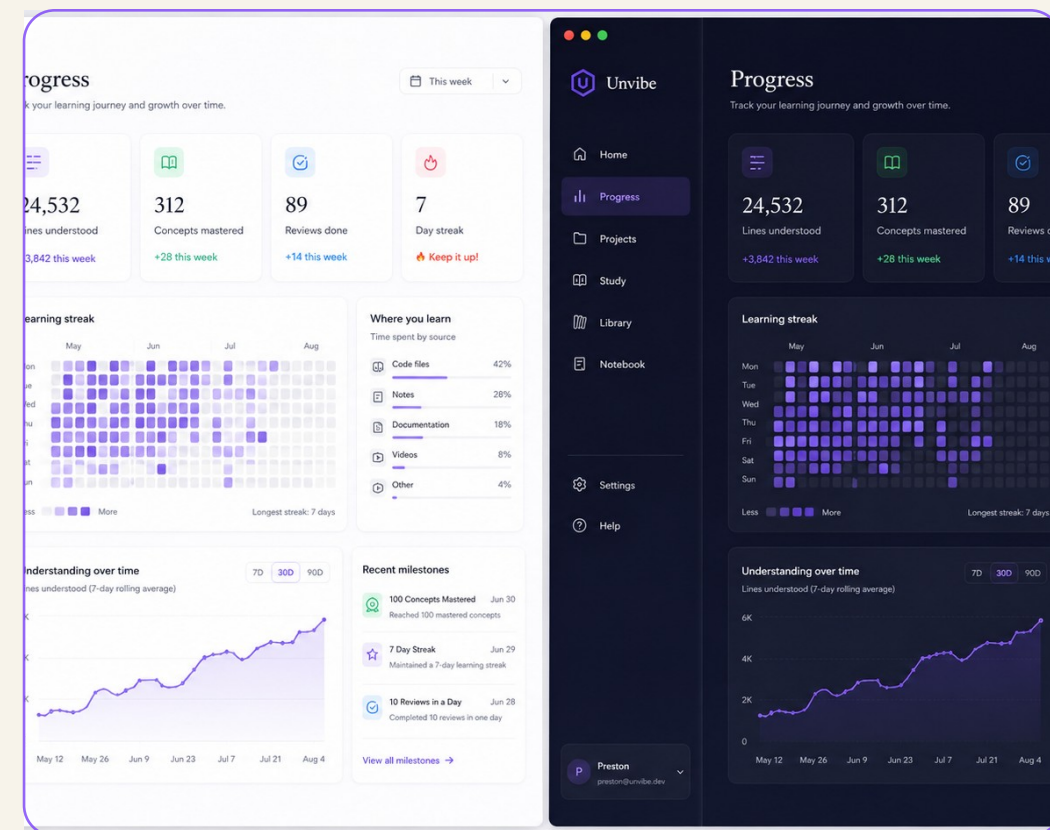
API routes

Strong

Async/await

Developing

Cautious labels, not fake mastery percentages.



The first users are people building faster than they understand.

Initial wedge: students and AI-assisted builders already using Cursor, Claude Code, GitHub, VS Code, and terminal agents.

 Students

Turn real projects into learning material

 AI-assisted builders

Understand what the agent changed

 Hackathon teams

Keep architecture straight while moving fast

 Beginner founders

Maintain and debug generated apps

Start free. Upgrade when projects grow.

The product scales from individual learning to team knowledge transfer without making Free useless.

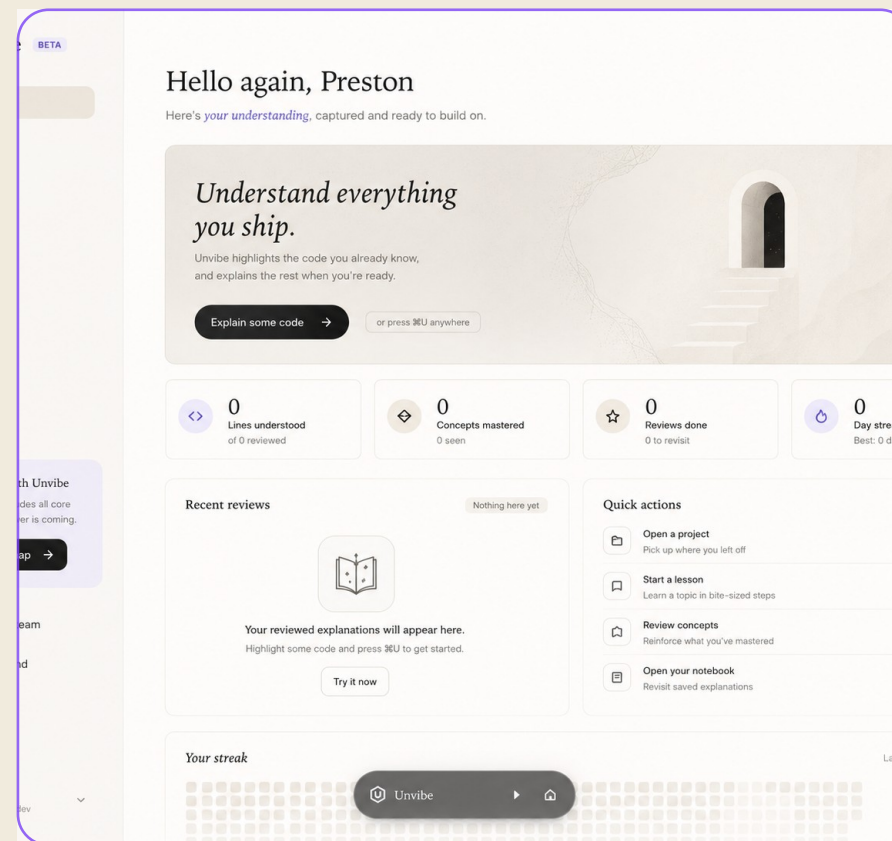
Free	Pro	Teams
\$0	\$8/mo	\$8/seat
30 explanations / month 1 active project Core overlay + saved learning	\$96/year = \$8/mo Larger repos + cross-file context Personal project memory	2-seat minimum Shared workspace \$6/seat/mo annual

Pricing plan mirrors the current Unvibe Free / Pro / Teams specification.

A community-led wedge around AI-assisted learning.

Unvibe can grow where the problem is already visible: students, hackathons, AI-builder communities, campus clubs, and dev tools audiences.

- 1 Private beta** Recruit active AI-coding users
- 2 Build-in-public demos** Show selected-code transformations
- 3 Campus + hackathons** Student ambassador loops
- 4 Content engine** Explain popular agent-generated patterns
- 5 Teams expansion** Onboarding and shared repo understanding



Adjacent tools generate, chat, or teach. Unvibe owns the learning layer.

The claim is not “no alternatives exist.” It is that no obvious product category owns code ownership inside the AI-assisted workflow.

Coding agents	Generate and edit code	Completion
General chatbots	Explain pasted code	Detached context
Courses	Teach fixed curriculum	Generic sequence
Unvibe	Teach from the code users build	Understanding + retention

What gets stronger over time: user-specific concepts, project memory, saved explanations, checks, preferences, and review patterns.

Private code requires a privacy-first product stance.

Unvibe should be trusted before it is scaled: intentional activation, local secret filtering, explicit context, account deletion, and honest beta status.

01 Selected context

On your Mac

02 Local secret filter

.env · tokens · keys

03 Approved request

Filtered context only

04 Saved learning

User-controlled

Current status: core reliability improved locally; external beta still needs staging verification, real-model tests, signing/notarization, and legal review.

Milestones to reach a credible private beta.

Core beta reliability	Auth, persistence, sync, cautious skills	Improved
Staging proof	Supabase RLS, real AI streaming, deletion	Next
Packaged Mac beta	Signing, notarization, clean install	Next
Retention evidence	Usage, weekly return, saved learning	Then
Teams foundation	Workspace model, seat rules, shared knowledge	Then

Capital accelerates a trusted learning layer around AI-generated software.

Funding supports the engineering and trust work required to turn a promising beta into a durable product category.

35% Product + engineering

20% AI infrastructure

15% Security + privacy + legal

15% Beta users + growth

10% Design + accessibility

5% Operations

Proposed ask: seed/pre-seed capital to reach credentialed private beta, paid-signal, Mac build, and retention evidence.

unvibe.site

Built to be compelling without pretending.

Unvibe should pitch the category with ambition, but keep product status, traction, security, billing, and integrations honest until verified.

No fabricated traction

Do not claim customers, revenue, partnerships, or downloads until verified.

Adjacent competitors exist

The differentiated wedge is learning ownership inside AI-assisted workflows.

Public status

Private beta / waitlist, unless a real downloadable build is live.

Market references

GitHub Octoverse 2025; IDC "Generative AI in the SDLC" paper.