

PRODUCT OVERVIEW

Repotrix AI

How it works - a plain-English tour

Documentation, diagrams, and answers that stay current with your code - automatically, on every commit, right inside your browser, your code editor, and your AI assistant. No prior technical background needed.

AI codebase intelligence

Web dashboard

VS Code

AI-assistant ready

What's inside

1	What is Repotrix?	The problem & the fix
2	The big picture	Web, editor & your AI assistant
3	Your journey, end to end	Sign-up -> answers
4	How Repotrix reads your code	Code -> knowledge
5	How chat works	Answers you can trust
6	Why it stays current	The freshness promise
7	Connect your AI assistant (MCP)	Repotrix inside your AI tool
8	The technology behind it	Built on proven tools
9	Every feature, in plain English	The surfaces
10	Security & trust	Your code stays safe
11	Plans	Start free, scale up
12	Why teams choose Repotrix	The short version
13	Glossary	Every term, decoded

HOW TO READ THIS

Look for the coloured boxes: GREEN = a plain-English analogy, CYAN = a key takeaway. Every technical term is also explained in the glossary at the back.

1

What is Repotrix?

The problem & the fix

Repotrix AI reads your software project, understands it, and keeps living documentation, diagrams, security reports and a chat assistant about it permanently up to date - automatically, every time your team changes the code.

The problem

Software is thousands of files of code. New people - or even the original authors months later - struggle to understand how it fits together. Teams write documentation, but the moment the code changes, the docs drift out of date. Within weeks nobody trusts them.

IN PLAIN ENGLISH

Imagine a giant Lego castle built by 20 people over two years, with a thick instruction booklet that nobody updates when bricks change. After a year it's useless and a new builder is lost. Repotrix is a robot that watches the castle and, every time a brick changes, instantly rewrites the matching page - so the instructions are always correct.

The fix

Most 'AI for your codebase' tools read your project once and answer from that frozen snapshot forever. Repotrix follows every change, figures out what changed, and regenerates only the affected docs, diagrams, security findings and search data. What you read is always what actually shipped.

THE ONE-SENTENCE PITCH

"Documentation, diagrams, and answers that stay current with your code - automatically, on every commit, right inside your browser, your code editor, and your AI assistant."

Who is it for?

- Developers who want to ask questions about a codebase and trust the answer.
- New joiners who need to get productive fast with onboarding guides built from the live code.
- Team leads & security folks who want diagrams, drift alerts and security scans that don't rot.

2

The big picture

Web, editor & your AI assistant - one brain

Repotrix meets you in three places - a web dashboard, a VS Code extension, and your own AI assistant - all powered by one shared backend (the 'brain').

The Backend

The brain

A secure server holding all the logic - accounts, the AI, reading repositories, storing everything. You never see it directly.

The Web app

repotrix.ai

The marketing site + the signed-in dashboard where you read docs, chat, view diagrams and manage your account.

VS Code extension

In your editor

Brings every Repotrix feature into VS Code, so developers never have to leave their code.

Your AI assistant

via MCP (paid)

Connect Repotrix to Claude Code, Cursor, Copilot, Gemini, ChatGPT or other MCP tools so the assistant you already use answers from your codebase.

IN PLAIN ENGLISH

Think of a restaurant. The backend is the kitchen - where the real cooking happens. The web app, the VS Code extension and your AI assistant are dining rooms where you place orders. All send orders to the same kitchen and get back the same food.



KEY TAKEAWAY

A feature is built once and instantly works in the website, the editor AND your AI assistant - so they always behave the same way, and improvements land everywhere at once.

3

Your journey, end to end

Following a new user, Asha

1. Step 1 - Sign up. Email + password, or one click via Google, GitHub, Bitbucket or Microsoft.
2. Step 2 - Connect a repository. Point Repotrix at your code; private code uses an access token, which is encrypted before it is stored.
3. Step 3 - Repotrix reads & understands it. A progress bar climbs while it reads the repo, removes any secrets, and builds a searchable understanding. A few minutes later it's ready.
4. Step 4 - Use the features. Chat, Documentation, Diagrams, Security, Onboarding, PR Reviews, Drift. Ask 'how does login work?' and get an answer with clickable citations.
5. Step 5 - Connect your AI assistant (paid). Link Cursor (or Claude Code) to Repotrix via MCP, so the AI tool itself can answer from your codebase.
6. Step 6 - The team keeps coding, and Repotrix keeps up. A teammate merges a change; Repotrix refreshes the affected docs and diagrams automatically. You never pressed 'regenerate'.

KEY TAKEAWAY

You only do two things by hand: connect a repo once, and ask questions. Everything else is automatic.

4

How Repotrix reads your code

Code -> searchable knowledge

Turning raw code into something the AI can search and explain happens in the background, so your experience stays fast. No secrets ever leave your control.

1. Read - Repotrix securely fetches a copy of your repository.
2. Protect - it removes any secrets or personal data before anything reaches the AI.
3. Understand - it breaks the code into pieces and turns each into a meaning-based 'fingerprint' (an embedding).
4. Index - those fingerprints are stored in a search built to find things by meaning, with each customer fully isolated.
5. Map - it builds a map of how files and functions depend on each other.
6. Generate - it produces the docs, diagrams, security report and onboarding guide.

IN PLAIN ENGLISH - WHAT'S A FINGERPRINT?

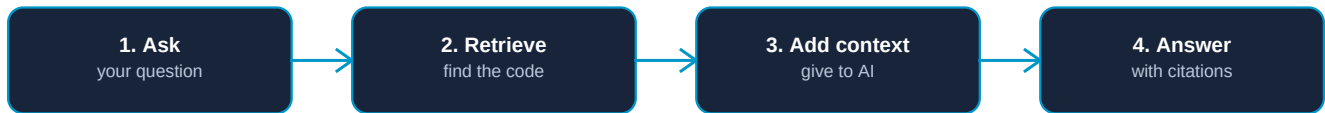
A 'fingerprint' (embedding) turns text into a point on a giant map of meaning. Code about 'user login' lands in one neighbourhood; 'billing' lands far away. Your question becomes a point too, and Repotrix finds the nearest code - that's how it searches by meaning, not keyword-matching.

5

How chat works

Accurate & citeable - RAG

The chat feature uses a well-proven technique called RAG (Retrieval-Augmented Generation). It exists to solve one problem: a raw AI doesn't know your private code, and you can't paste a million lines into it.



Repotrix finds the most relevant code from your current commit, hands it to the AI, and the AI answers using only that - streamed back live, with citations to the exact files.

IN PLAIN ENGLISH

It's an open-book exam. The AI isn't trying to remember your code from memory (it can't - it's private). Repotrix flips to the right pages, lays them on the desk, and the AI answers using only what's on the desk. That's why answers stay accurate and can cite their sources.

6

Why it stays current

The freshness promise

This is the heart of Repotrix. Other tools generate documentation once; Repotrix keeps it continuously in step with your code.

1. When you connect a repository, your code host notifies Repotrix automatically on every push.
2. Each notification is cryptographically verified, so Repotrix knows it's genuine.
3. Repotrix sees exactly which files changed and refreshes only the affected docs, diagrams, security and search.
4. Every artifact records the exact commit it was built from, so you can always see whether something is current, updating, or being rebuilt.

IN PLAIN ENGLISH

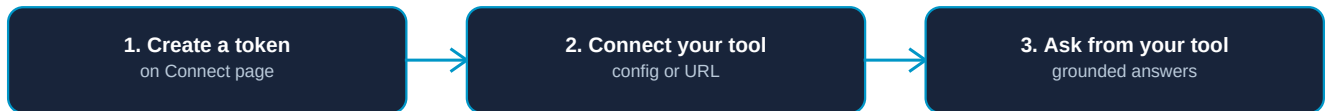
A normal docs tool is a photograph: accurate the day it's taken, slowly wrong forever after. Repotrix is a live mirror: it always shows your code as it is right now. It even sends a weekly summary of everything it quietly kept tidy for you.

7

Connect your AI assistant (MCP)

Repotrix inside your own AI tool

On paid plans you can use Repotrix from inside your own AI tool - Claude Code, Cursor, Copilot, Gemini, ChatGPT and other MCP-compatible tools - over the Model Context Protocol (MCP), an open standard for connecting AI tools to outside services. Your assistant can then answer from your live codebase without you copy-pasting anything.



Editor tools (Claude Code, Cursor, Copilot, Gemini CLI) add a small config snippet; URL-based tools (ChatGPT, Gemini) take a server URL with your token. The Connect page in Repotrix gives you the exact setup for each.

What your assistant can do

- List your repositories.
- Ask a question about a repo and get a grounded, cited answer.
- Search your code by meaning to find the right files.
- Fetch the architecture overview, and see what recently changed.

KEY TAKEAWAY

Same grounded, always-current answers as the web app and the extension - now inside the AI tool you already code in. Available on paid plans.

8

The technology behind it

Built on proven, industry-standard tools

The backend (the brain)

Technology	What it is	How Repotrix uses it
Python & FastAPI	The language of AI + a fast web framework	Runs all the logic; serves every surface
PostgreSQL	A rock-solid database	Stores accounts, repos, jobs and billing
Redis	A lightning-fast in-memory store	Manages the queue of background jobs
Pinecone	A vector database (finds by meaning)	Powers search & chat; each customer isolated
Google Gemini	State-of-the-art AI models	Writes docs/answers; creates the embeddings
Encryption & OAuth	Security building blocks	Protects tokens; enables one-click sign-in

The web app, extension & AI bridge

Technology	What it is	How Repotrix uses it
TypeScript	A safer version of the web's language	Powers the website, the extension and the bridge
Next.js / React	The leading website framework	Builds the marketing site and dashboard
Mermaid	Turns text into diagrams	Renders the architecture / dependency / data-flow diagrams
VS Code API	The official way to extend VS Code	Adds the Repotrix sidebar and panels
MCP	Open standard for AI-tool plugins	Lets Claude, Cursor, Copilot, Gemini, ChatGPT call Repotrix

KEY TAKEAWAY

Nothing here is exotic - Repotrix combines proven, well-supported technologies, each chosen as the best fit for its job.

9

Every feature, in plain English

What each surface does for you

Feature	What it does for you
Chat	Ask anything; an answer grounded in the current commit, with citations to the exact files.
Documentation	Full technical docs written from your code, rewritten when a merge changes them. Export PDF/Markdown.
Architecture diagrams	Architecture, dependency and data-flow diagrams that regenerate on every push. Export SVG/PNG.
Security reports	Scans for leaked secrets, personal data and licensing issues, with a severity-ranked health score.
Onboarding guides	Role-aware 'getting started' guides built from the live codebase.
PR reviews	Every pull request reviewed against your current docs; findings posted onto the PR automatically.
Drift detection	Flags the moment your real architecture diverges from its documentation.
Health analyzer	Live metrics on code quality, complexity and maintainability.
Connect AI (MCP)	Use all of the above from inside Claude Code, Cursor, Copilot, Gemini, ChatGPT and other MCP-compatible tools. Paid plans.

IN PLAIN ENGLISH

It's a smart, always-awake teammate who has read every line, never forgets, re-reads the instant anything changes, and is happy to explain any part, draw you a map, flag anything dangerous, or onboard the new person - without ever being annoyed.

10

Security & trust

Your code stays safe

Repotrix reads private source code, so trust is everything. Security is layered throughout.

Protecting your account

- Passwords are never stored as text - only as a one-way scrambled 'hash'.
- Your sign-in is kept in a secure, script-proof store; tokens expire and die on logout.
- One-click sign-in via Google, GitHub, Bitbucket and Microsoft lets you avoid passwords entirely.

Protecting your code & secrets

- Secrets removed before the AI: any API keys, passwords or personal data are replaced with [REDACTED] before a single byte reaches the AI model.
- Encryption at rest: your access token and AI key are encrypted - a database leak alone wouldn't expose them.
- Full isolation: each customer's data lives in its own space; one account can never read another's.

Protecting the platform

- Verified notifications - change-pings are cryptographically checked for authenticity.
- Rate limiting & audit logging - abuse is blocked and sensitive actions are recorded.
- Encrypted everywhere, with nightly backups that are regularly restore-tested.

KEY TAKEAWAY

Security here isn't one feature - it's a habit at every layer: hashed passwords, encrypted secrets, code scrubbed before the AI, isolated customers, and verified notifications.

11

Plans

Start free, scale up

Repotrix comes in Free, Paid and Enterprise tiers, so you can start free and grow as your team does. Paid is metered by the number of repositories. The AI-assistant (MCP) integration is available on paid plans.

Free**Paid****Enterprise****[SEE FULL DETAILS](#)**

Sign in - or create a free account - at repotrix.ai to see full plan details, limits and pricing.

REPOTRIX

12

Why teams choose Repotrix

The short version

1. Documentation you can actually trust - it stays current on every commit.
2. Answers grounded in your real code - chat cites the exact files; no confident guesses.
3. Wherever you work - web dashboard, VS Code extension, and your own AI assistant, all in sync.
4. Faster onboarding - new hires learn from guides built from the live codebase, not a stale wiki.
5. Built-in safety net - security scans, drift alerts and PR reviews catch problems early.
6. You stay in control - bring your own AI key, manage your tokens, keep data isolated & encrypted.

THE SHORT VERSION

Repotrix turns your codebase into a living, trustworthy knowledge base that explains itself, keeps itself current, and meets your team in the browser, the editor, and the AI assistant.

REPOTRIX

13

Glossary

Every term, decoded

Term	Means...
Repository (repo)	A folder with all the files of a software project, plus its change history.
Commit / push	A saved change to the code (commit), sent to the shared server (push).
Backend / Frontend	The hidden server that does the work / the part you see and click.
Embedding	Text turned into numbers capturing its meaning, so similar ideas sit 'near' each other.
LLM	Large Language Model - the AI that reads and writes human-like text (here, Google Gemini).
RAG	Retrieval-Augmented Generation - fetch the relevant code first, then let the AI answer using it.
Vector database	A database that finds things by meaning rather than exact words.
Webhook	An automatic 'ping' one service sends another when something happens (e.g. 'code was pushed').
MCP	Model Context Protocol - an open standard that lets AI tools call services like Repotrix.
Access token (PAT)	A secret key that lets your AI tool talk to Repotrix on your behalf.
OAuth	Signing in to one app using your account from another (Google/GitHub), without sharing a password.
Encryption at rest	Stored data is locked so a leak alone can't reveal it.
Drift	When the real code and its documentation have quietly diverged.
SaaS	Software as a Service - software you subscribe to online instead of installing.
Citation	A link in an answer pointing to the exact file the answer came from.