

# DocGen — Automatic Documentation Generator

AI-driven agent that analyzes Python codebases and produces structured, readable Markdown documentation. Presented by Uplakshya Tiwari.





## Problem: Documentation Lags Behind Code

### Time-consuming upkeep

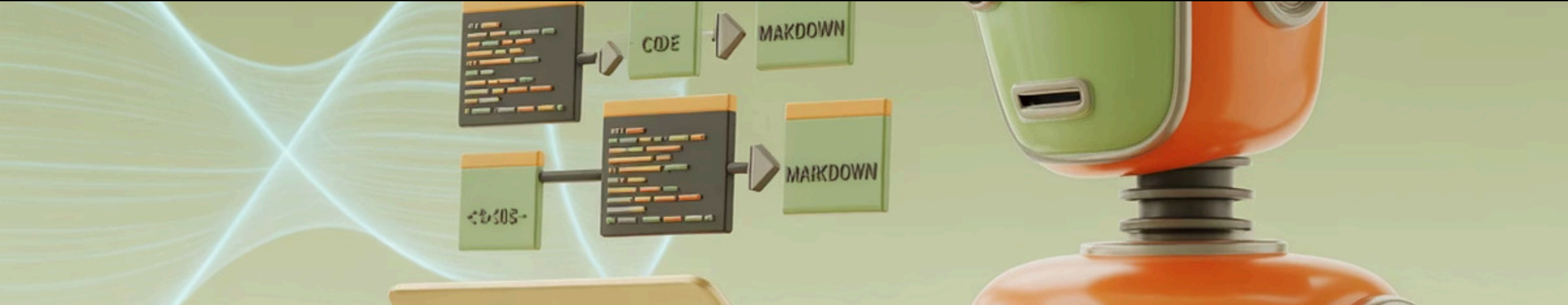
Maintaining docs requires manual effort that teams rarely prioritize.

### Docs drift stale

Documentation often becomes inaccurate as code evolves.

### Human error

Manual summaries miss edge cases and implementation details.



# Proposed Solution: DocGen

DocGen automatically scans a Python project, extracts functions and classes, and generates clean Markdown documentation with structured sections and examples.

## Scan

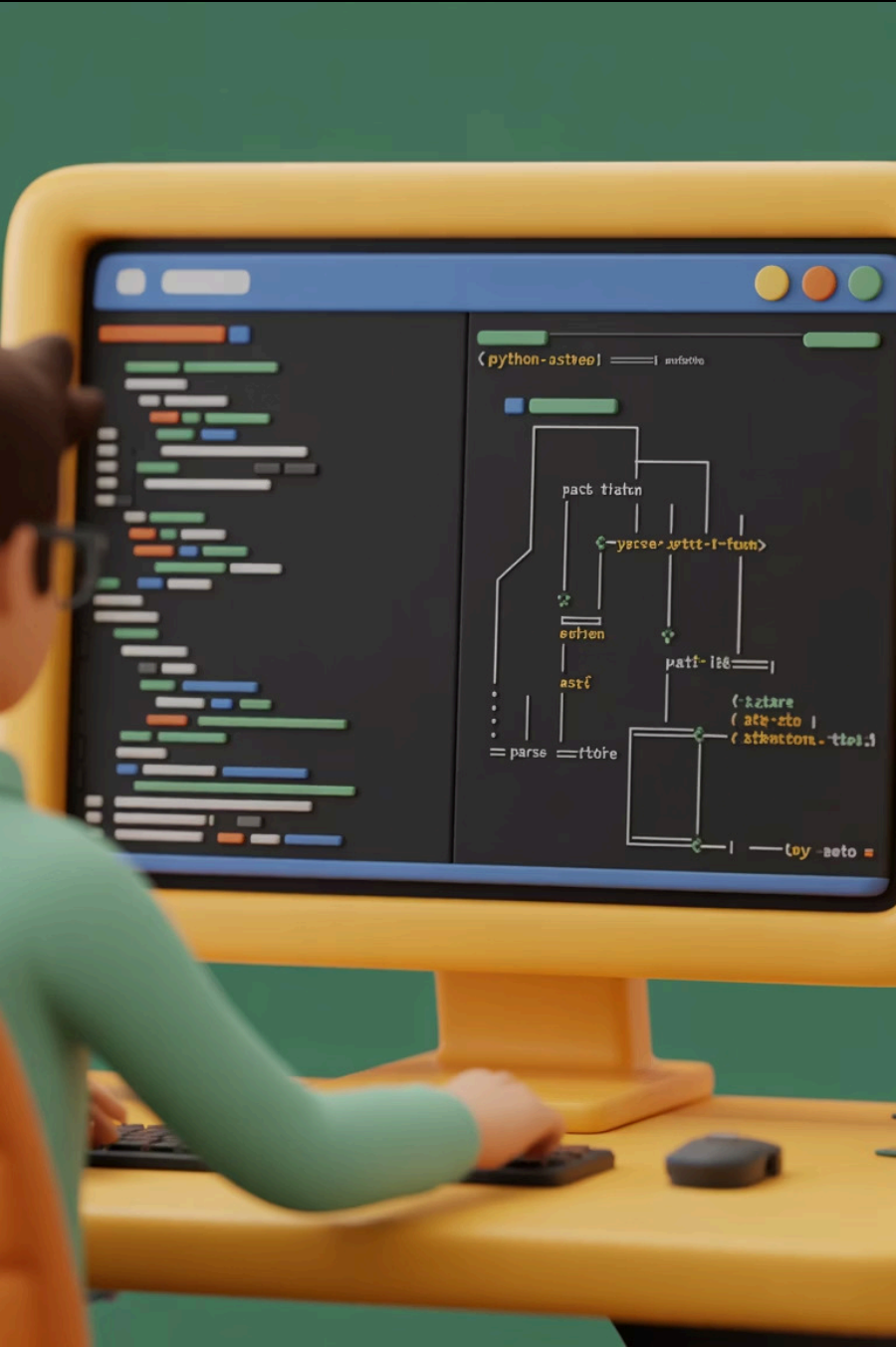
Traverse repository to locate source files and modules.

## Extract

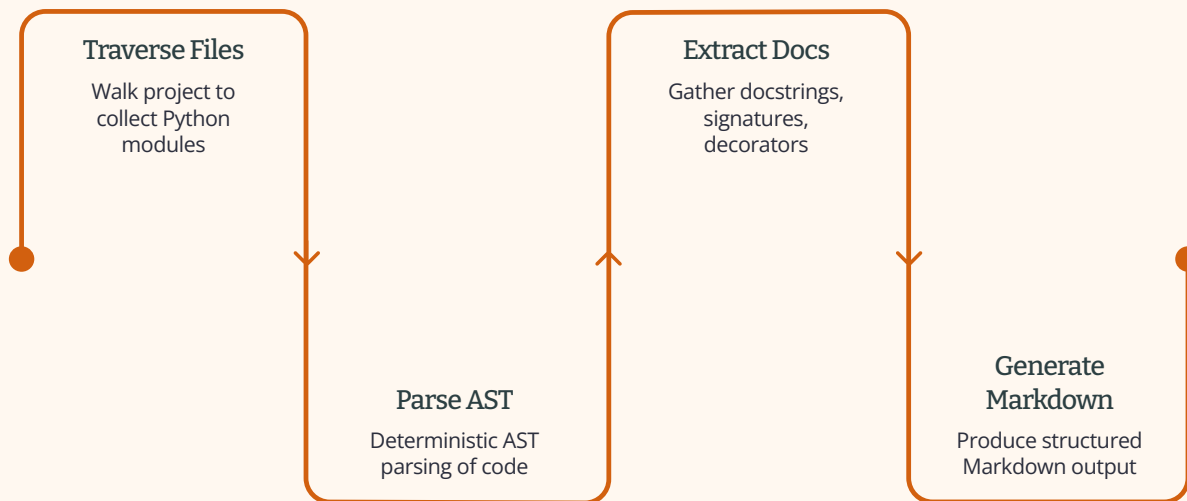
Parse AST to retrieve defs, signatures, and docstrings.

## Emit

Render well-formatted Markdown (README, DOCUMENTATION.md, API sections).



## How It Works — Technical Flow



DocGen performs deterministic AST parsing to avoid brittle regex parsing, preserving type hints, decorators, and nested scopes.



## Agent Behavior

- Goal-driven: focuses on producing documentation artifacts, not conversational replies.
- Autonomous: executes pipeline end-to-end with minimal human supervision.
- Input: code folder (repo or local directory).
- Output: structured Markdown files and a DOCUMENTATION.md landing page.

📄 Designed for CI integration — run on push, PR, or scheduled jobs.



# Tech Stack



## Python

Core language for parsing, orchestration, and templates.



## AST module

Reliable, structured parsing of functions, classes, and signatures.



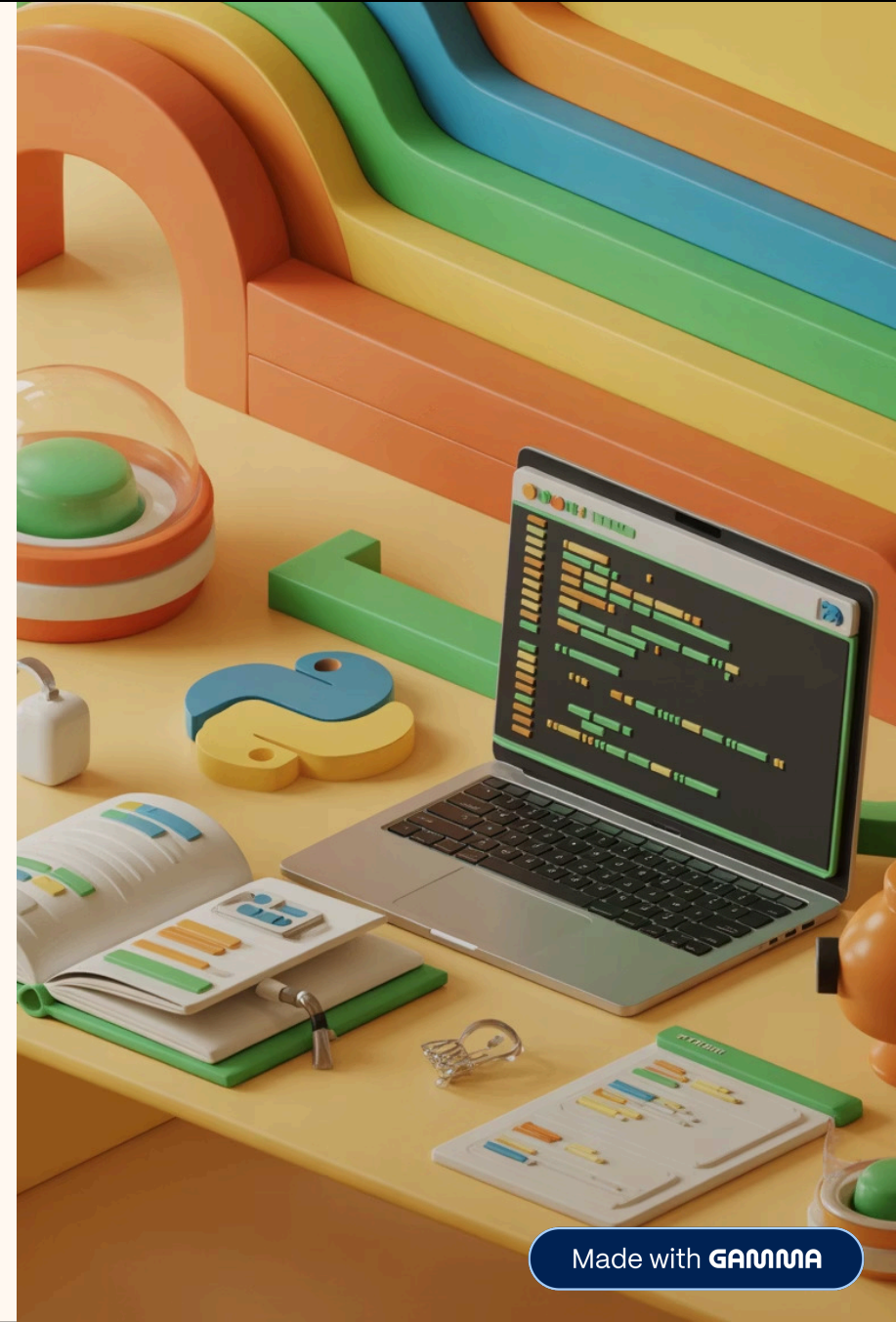
## Markdown

Plain, portable output for READMEs, API docs, and examples.



## GitHub

Repo integration, CI/CD pipelines, and hosting for generated docs.



# Project Structure (Example)



src/

└─ docgen.py — core pipeline and CLI

sample\_project/ — targets for parsing and test coverage

requirements.txt — reproducible environment

README.md — repo overview and usage



docgen.py exposes a CLI and a library API for integration into tests or CI.

# Results

- **Generated DOCUMENTATION.md**

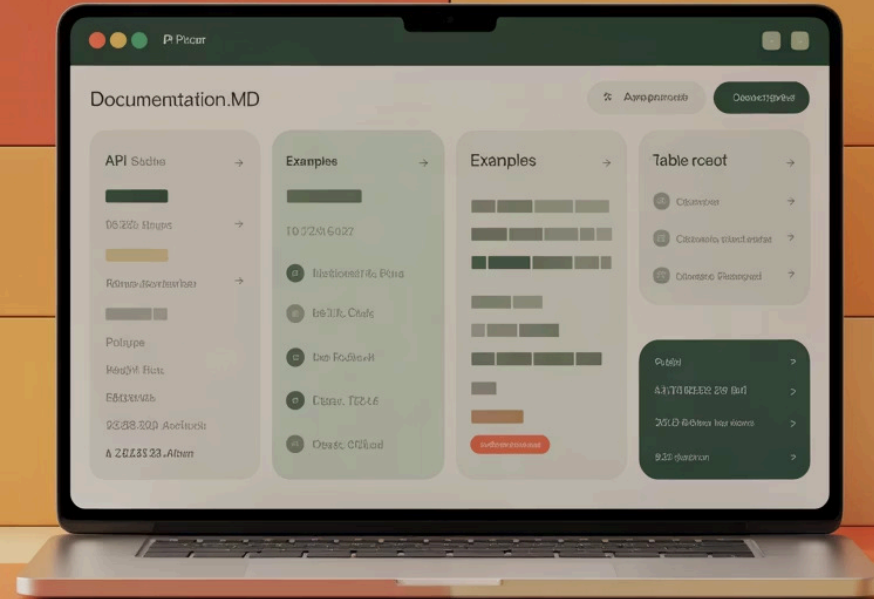
Comprehensive landing doc with Table of Contents, module summaries, and examples.

- **Reduced manual effort**

Teams save time on routine updates and focus on higher-value tasks.

- **Hackathon-ready**

Lightweight, reliable tool suitable for student projects and prototypes.





# Future Scope

## Multi-language support

Extend parsers to JS/TS, Java, and Go with pluggable front-ends.

## API documentation

Generate OpenAPI-compatible schemas and endpoint docs from code and decorators.

## UI dashboard

Interactive viewer for docs with search, diffing, and live previews.





# Thank You

## Thank You

GitHub: Documentation-Generator-Agent

Open to feedback and collaboration — happy to demo the pipeline and walk through code samples.



REPO

FEEDBACK WELCOME