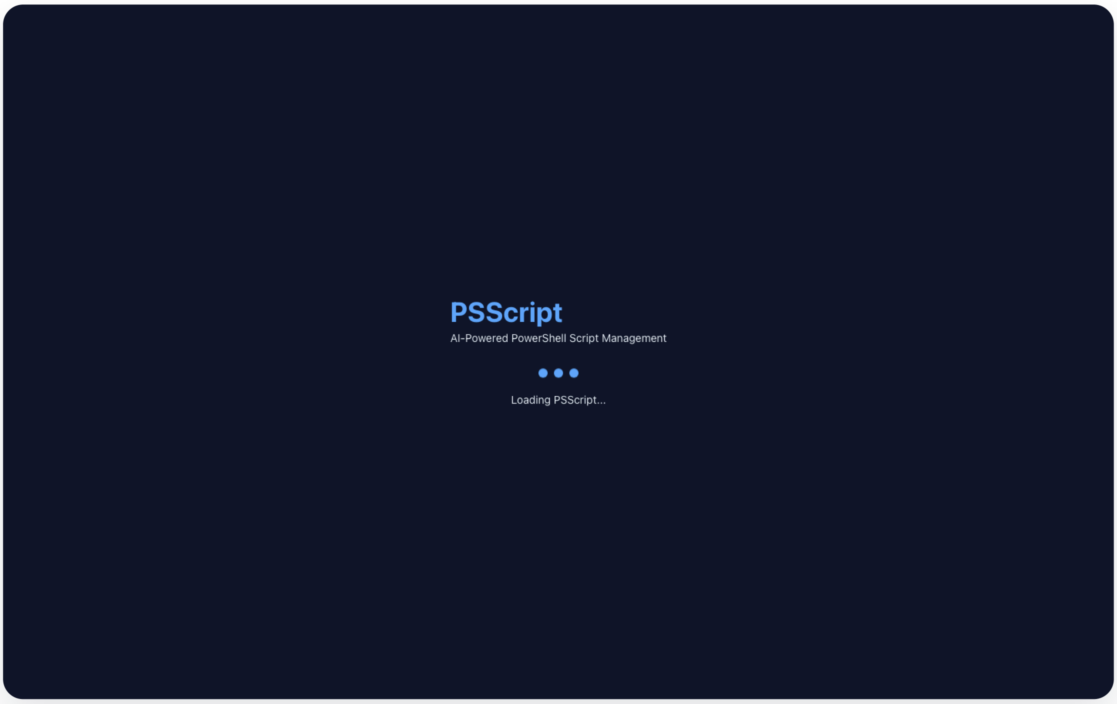
PSScript Manager

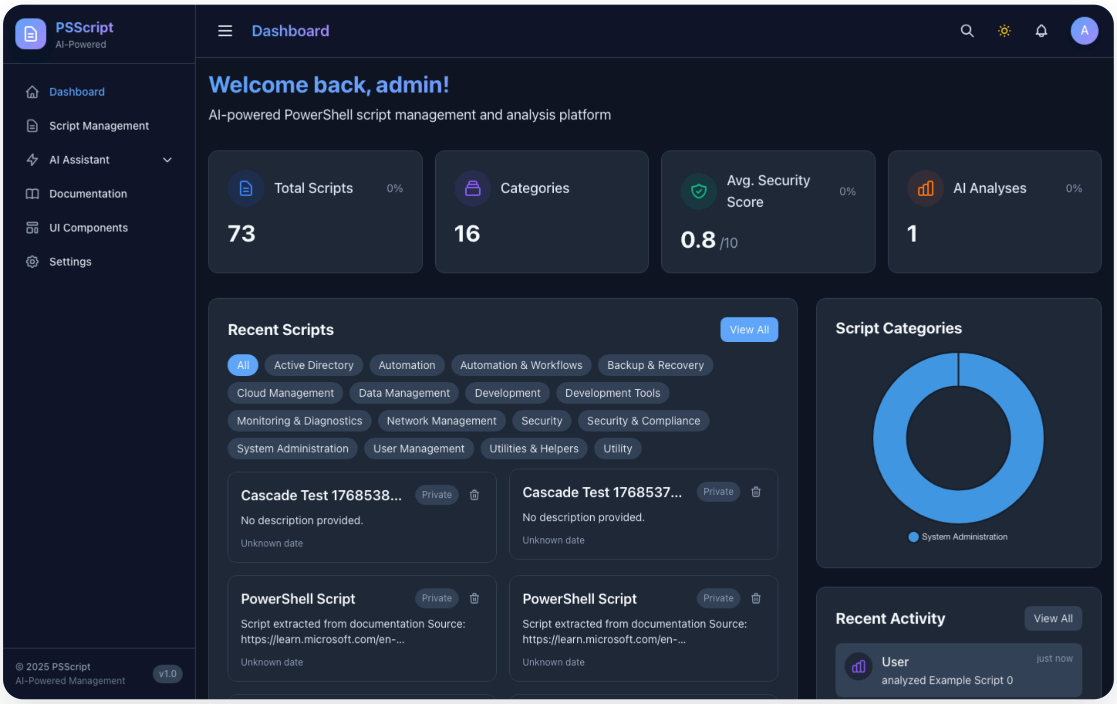
AI-powered PowerShell script management: intake, analysis, discovery, agentic chat, and governance—presented in the same polished style used by top open-source GitHub READMEs (think Supabase/PostHog).

Quick tour (screenshots)

- Login (demo-ready):



- Dashboard (stats, trends, activity):



Welcome back, admin!

AI-powered PowerShell script management and analysis platform

Total Scripts 0% 73

Categories 16

Avg. Security Score 0% 0.8 /10

AI Analyses 0% 1

Cascade Test 1768538... Private
No description provided.
Unknown date

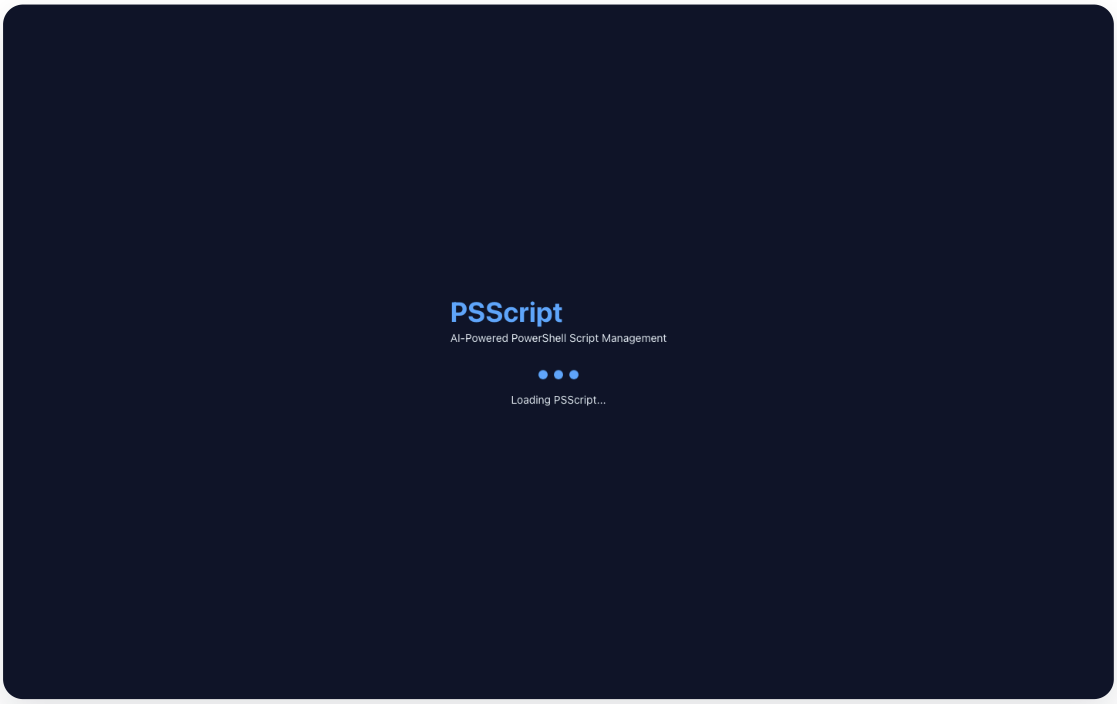
Cascade Test 1768537... Private
No description provided.
Unknown date

PowerShell Script Private
Script extracted from documentation Source: https://learn.microsoft.com/en-...
Unknown date

PowerShell Script Private
Script extracted from documentation Source: https://learn.microsoft.com/en-...
Unknown date

User just now
analyzed Example Script 0

- Script library (categories, owners, versions):

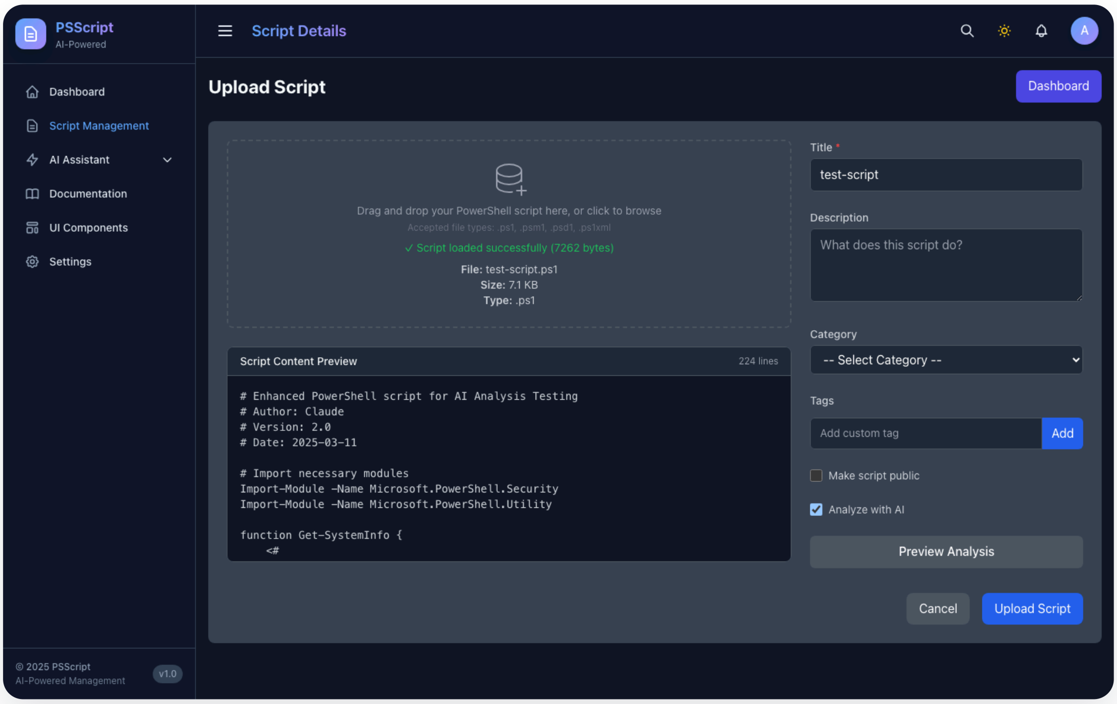


PSScript

AI-Powered PowerShell Script Management

Loading PSScript...

- Upload with live preview + dedup hints:



Upload Script

Drag and drop your PowerShell script here, or click to browse

Accepted file types: .ps1, .psm1, .psd1, .ps1xml

✓ Script loaded successfully (7262 bytes)

File: test-script.ps1
Size: 7.1 KB
Type: .ps1

Script Content Preview 224 lines

```
# Enhanced PowerShell script for AI Analysis Testing
# Author: Claude
# Version: 2.0
# Date: 2025-03-11

# Import necessary modules
Import-Module -Name Microsoft.PowerShell.Security
Import-Module -Name Microsoft.PowerShell.Utility

function Get-SystemInfo {
    <#
```

Title *: test-script

Description: What does this script do?

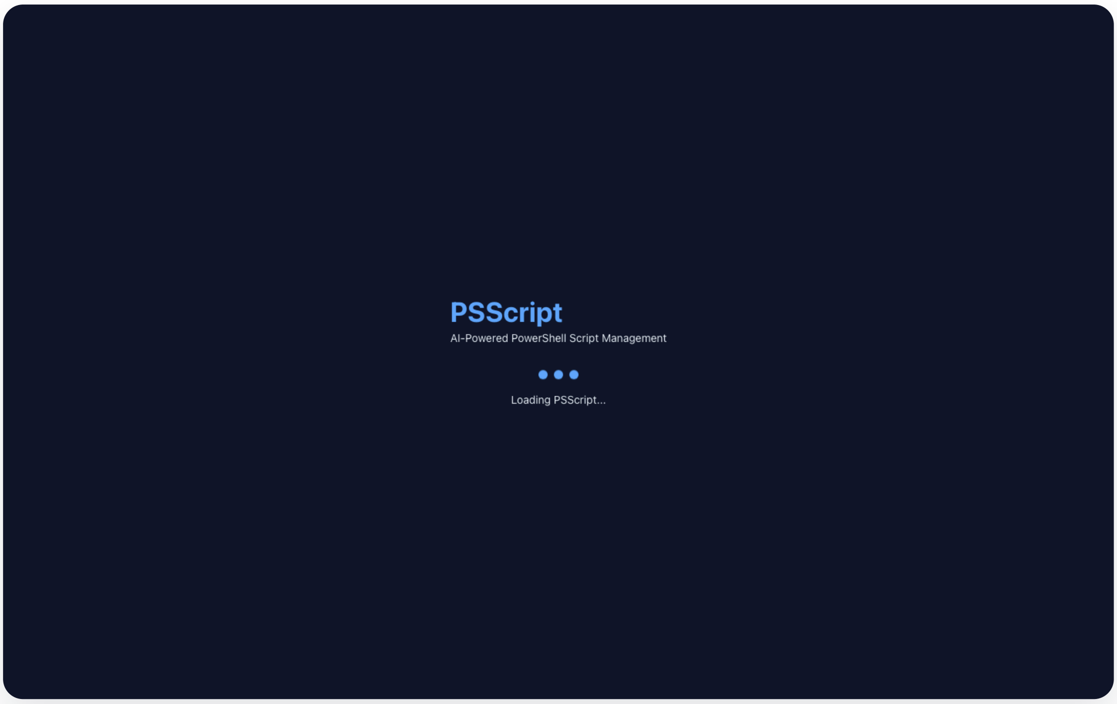
Category: -- Select Category --

Tags: Add custom tag

□ Make script public

☑ Analyze with AI

- AI analysis and remediation:

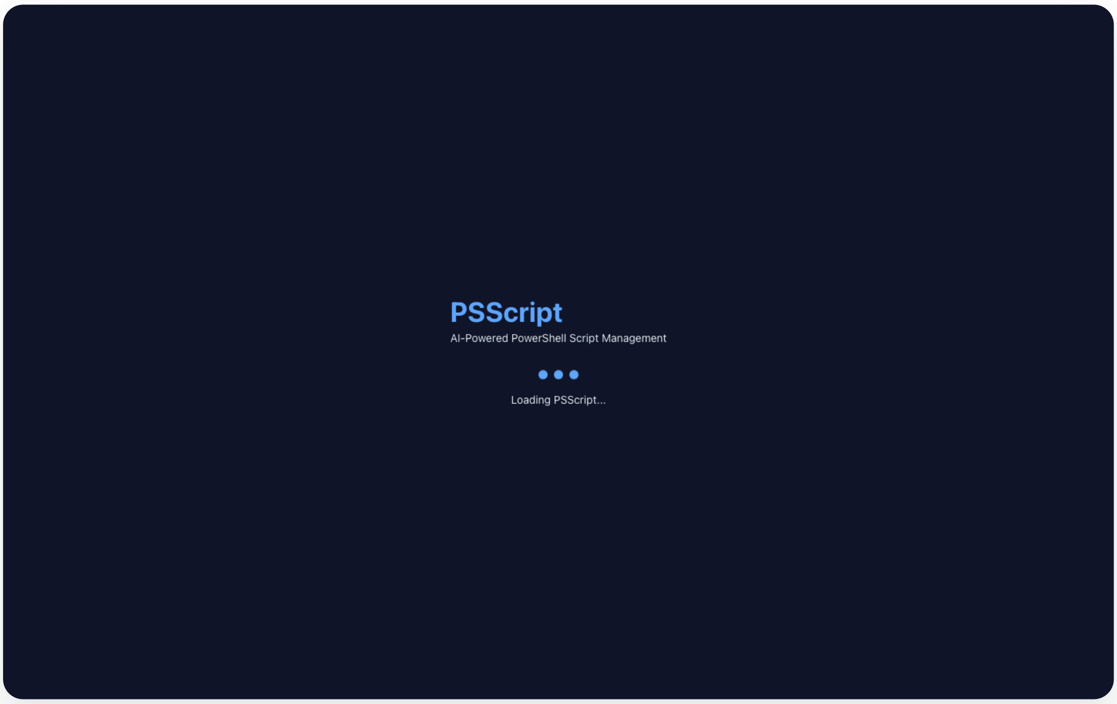


PSScript

AI-Powered PowerShell Script Management

Loading PSScript...

- Documentation explorer and crawler:

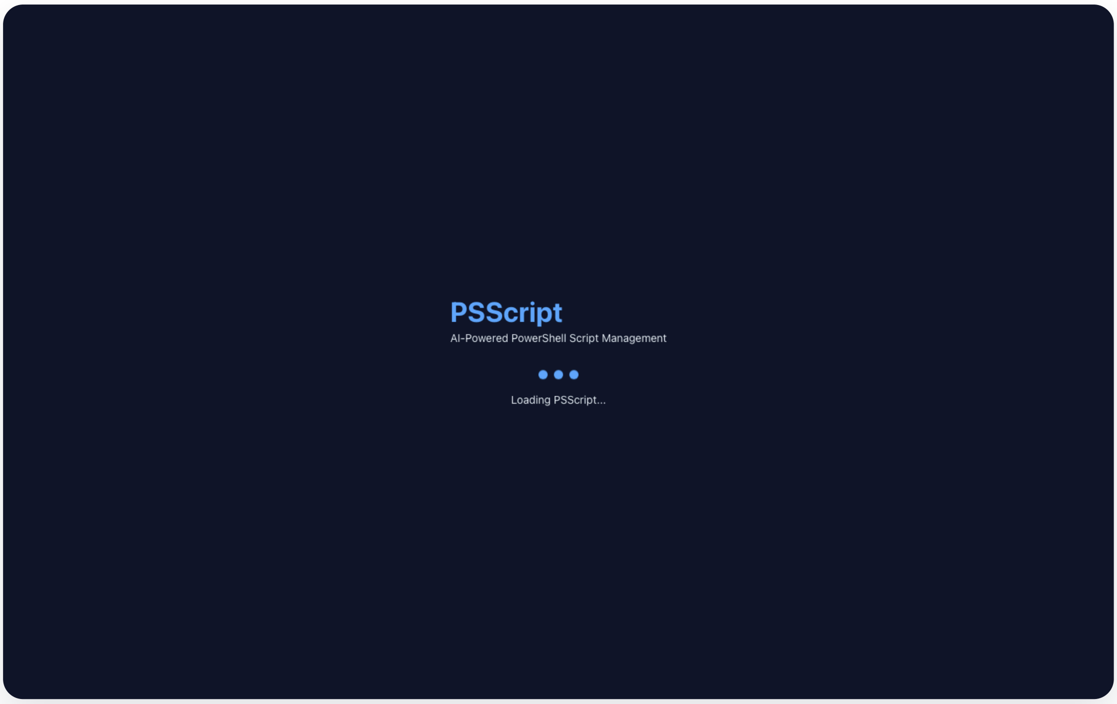


PSScript

AI-Powered PowerShell Script Management

Loading PSScript...

- Chat + agentic workflows:

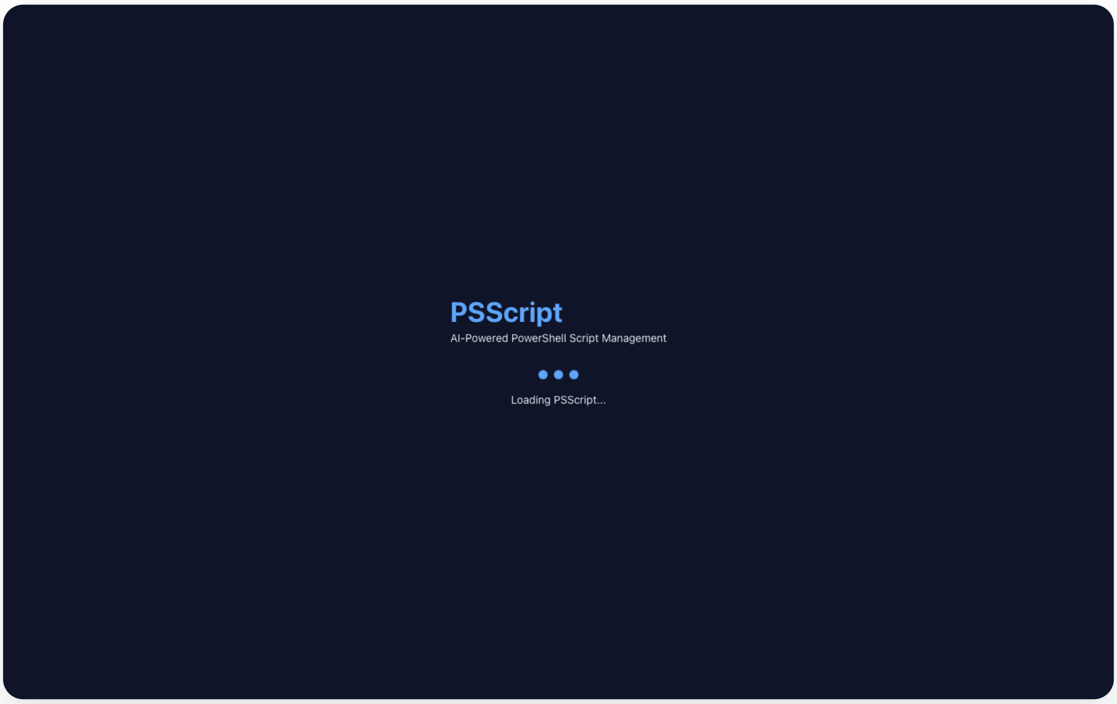


PSScript

AI-Powered PowerShell Script Management

Loading PSScript...

- Analytics and governance KPIs:

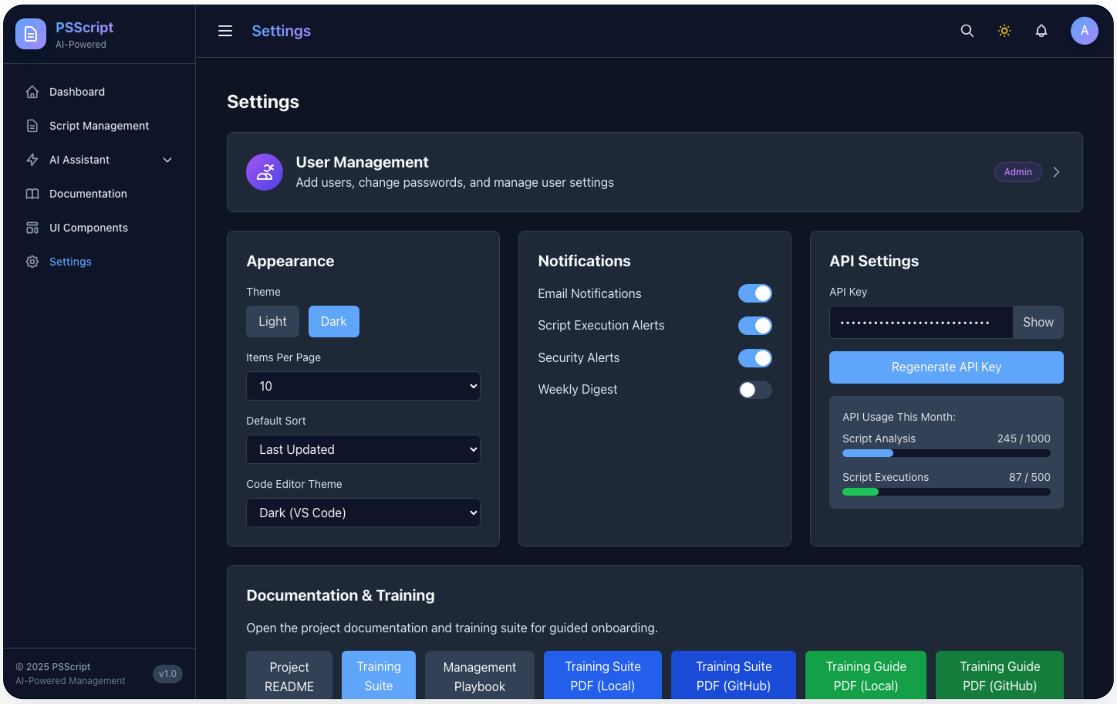


PSScript

AI-Powered PowerShell Script Management

Loading PSScript...

- Settings with training/exports links:



Settings

User Management

Add users, change passwords, and manage user settings

Appearance

Theme: Light [✓ Dark]

Items Per Page: 10

Default Sort: Last Updated

Code Editor Theme: Dark (VS Code)

Notifications

Email Notifications ●

Script Execution Alerts ●

Security Alerts ●

Weekly Digest ○

API Settings

API Key: ••••••••••••••••••••••••••••

API Usage This Month:

Script Analysis	245 / 1000
Script Executions	87 / 500

Documentation & Training

Open the project documentation and training suite for guided onboarding.

Project README
Training Suite
Management Playbook
Training Suite PDF (Local)
Training Suite PDF (GitHub)
Training Guide PDF (Local)
Training Guide PDF (GitHub)

Product highlights

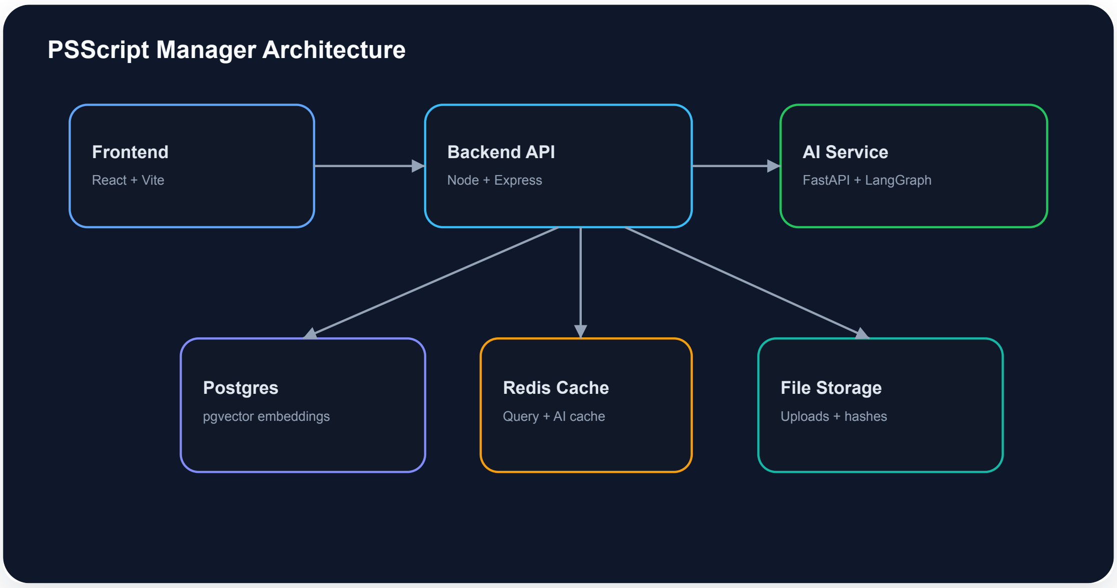
- [x] Script intake: tagging, version history, hash-based deduplication
- [x] AI analysis: security scores, findings, remediation notes, audit trails
- [x] Discovery: keyword + vector search, documentation explorer, crawler
- [x] AI copilots: chat assistant, agent orchestration, history
- [x] Operations: analytics dashboards, exports (HTML/PDF/DOCX), health checks
- [x] Training suite: modules, labs, role paths, printable guides (linked in Settings)

System at a glance

Layer	What it does	Tech
UI	React + Tailwind, Monaco editor, dark/light theming	Vite, React 18, Tailwind
API	Script CRUD, analysis orchestration, auth, docs crawl	Node.js/Express, Sequelize
AI	Analysis scoring, recommendations, embeddings	FastAPI, pgvector
Data	Postgres (pgvector), Redis cache	Dockerized services
Ops	Cloudflare tunnel, health checks, logs, exports	Docker Compose

Architecture and flows:

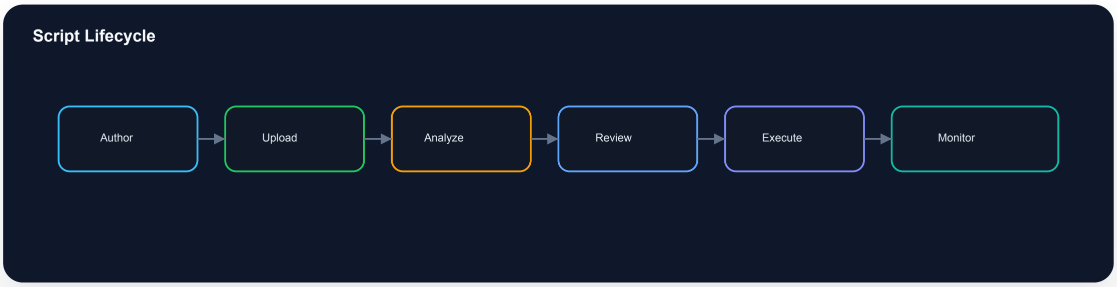
- Architecture:



- Analysis pipeline:

Analysis Pipeline

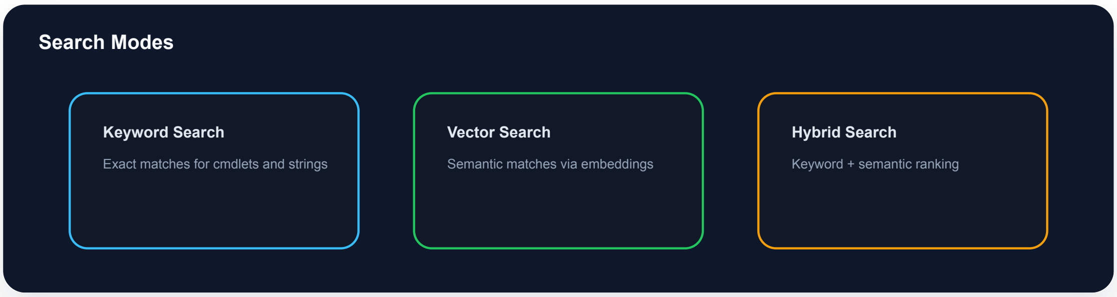
- Script lifecycle:



Script Lifecycle

Author → Upload → Analyze → Review → Execute → Monitor

- Search modes:



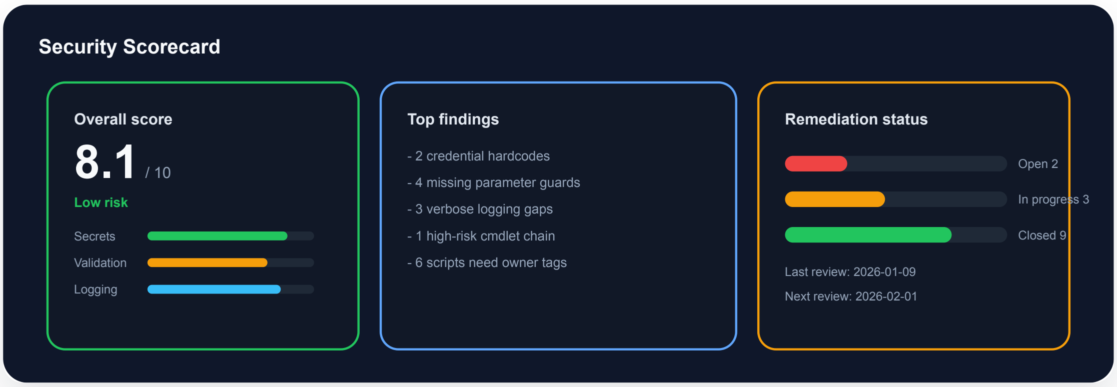
Search Modes

Keyword Search
Exact matches for cmdlets and strings

Vector Search
Semantic matches via embeddings

Hybrid Search
Keyword + semantic ranking

- Security scorecard:



Security Scorecard

Overall score

8.1 / 10

Low risk

Secrets
Validation
Logging

Top findings

- 2 credential hardcodes
- 4 missing parameter guards
- 3 verbose logging gaps
- 1 high-risk cmdlet chain
- 6 scripts need owner tags

Remediation status

Open 2
In progress 3
Closed 9

Last review: 2026-01-09
Next review: 2026-02-01

Getting started

Fast mock start (best for demos/training)

```
./start-all-mock.sh
# Frontend: http://localhost:3002
# Backend API: http://localhost:4000/api
# AI service: http://localhost:8000
```

Production Docker

```
./docker-deploy.sh
# Frontend: http://localhost:3002
# Backend API: http://localhost:4000
```

See `DOCKER-QUICKSTART.md` and `DOCKER-SETUP.md` for details.

Local development

```
npm run install:all
npm run dev
```

Playwright screenshots (used in this README)

```
./scripts/capture-readme-screenshots.sh
# Captures to docs/screenshots/*
```

Training suite (supreme docs pack)

- Location: `docs/training-suite/`
- Contents: modules, labs, guided walkthrough, schedules, rubrics, screenshots
- In-app: Settings → Documentation & Training → Training Suite / PDFs / DOCX
- Exports: `scripts/export-docs.sh --all` → `docs/exports/{html,pdf,docx}/`
- `docs/exports/pdf/Training-Suite.pdf`
- `docs/exports/docx/Training-Suite.docx`

Key workflows

Flow	Steps	Where
Upload & dedup	Upload → tags → hash check → version history	Scripts → Upload
AI review	Analyze → scores → findings → remediation notes	Scripts → Analysis
Discovery	Keyword + vector → filters → detail → history	Scripts / Documentation
Chat/agents	Ask for fixes, schedules, safety → save history	Chat / AI Assistant
Ops & governance	Analytics → scorecards → exports → health	Analytics / Settings

Environment & config

- `.env` samples: `src/frontend/.env`, `src/backend/.env`
- Mock mode toggles: `VITE_USE_MOCKS=true`, `USE_MOCK_SERVICES=true`
- Ports: UI 3002, API 4000, AI 8000 (mock defaults)
- CSRF/CORS dev origins: localhost:3000/3002 + 4000 (see `src/backend/src/middleware/security.ts`)

Contributing and licenses

- Contributions welcome: see `CONTRIBUTING.md`
- License: MIT (`LICENSE`)

Generated 2026-01-16 23:34 UTC