

Training Guide

End-to-end walkthrough with screenshots and labs

PSScript Manager Training

PSScript Training Guide

End-to-end walkthrough of the PowerShell Script Manager. Includes flows, screenshots, labs, tables, sample actions, and operational checklists. Built for hands-on enablement and printable exports.

Table of Contents

- Program overview
- System overview
- Feature cheat sheet
- Training roadmap
- Environment setup
- Guided walkthrough
- Role-based paths
- Module breakdown
- Hands-on labs
- Sample actions
- Operational readiness
- Appendix

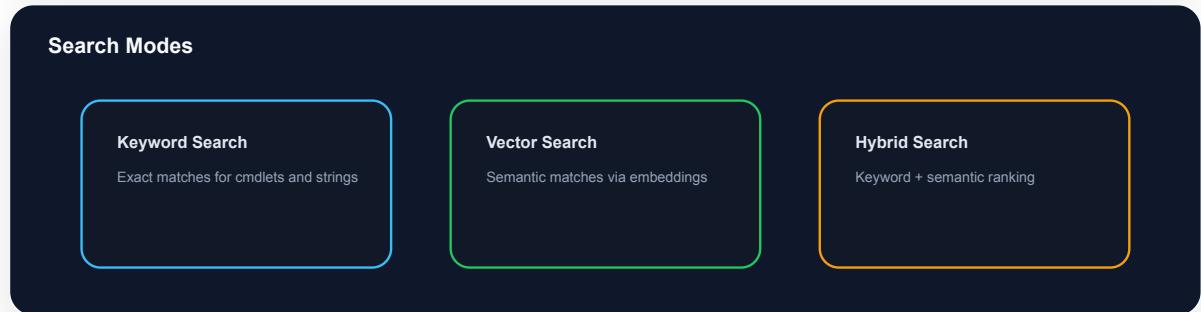
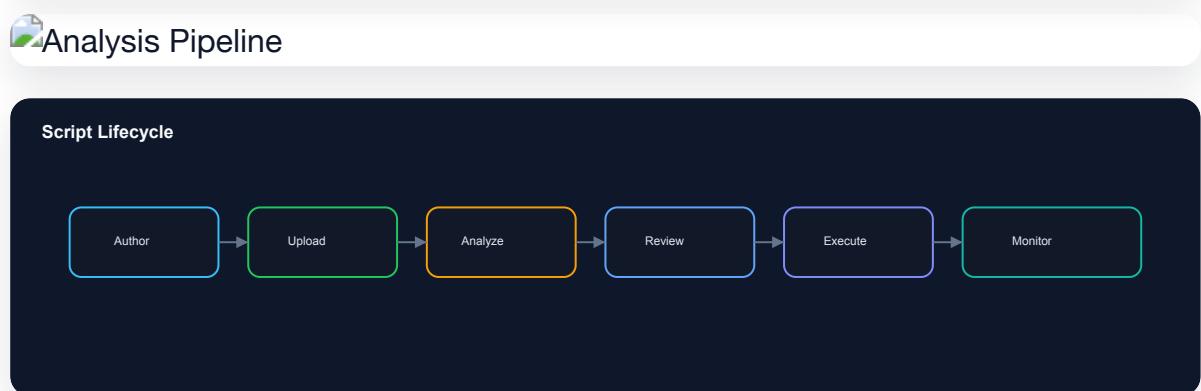
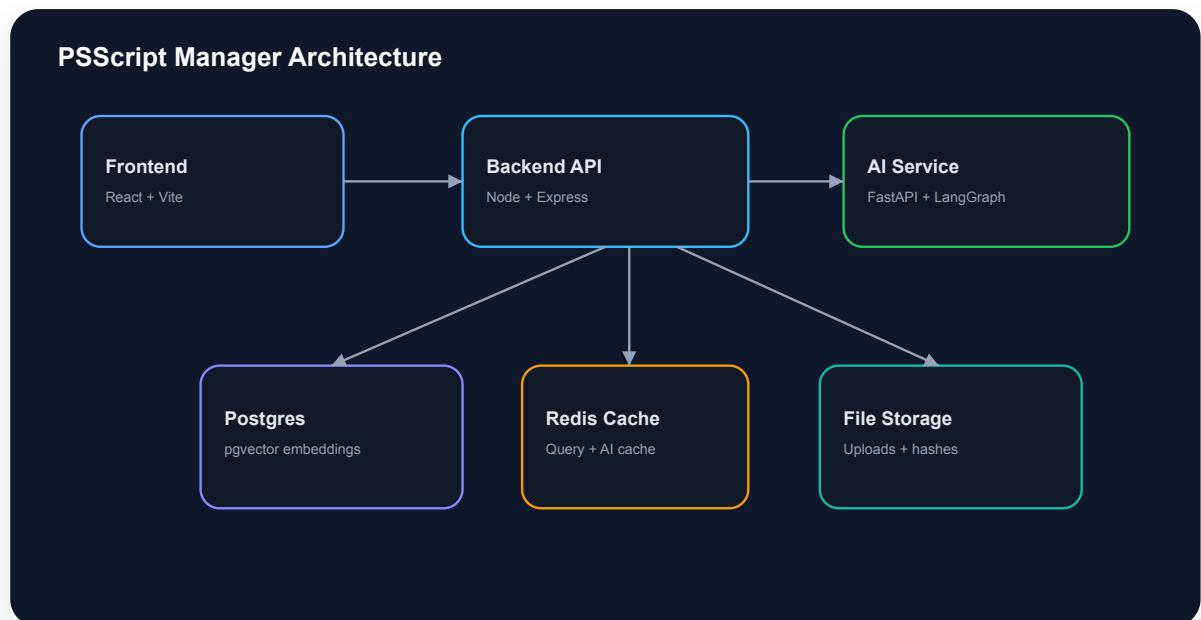
Program overview

PSScript Manager is a full workflow platform for PowerShell scripts—intake, AI analysis, discovery, agentic chat, and governance. This guide delivers:

- Organized, tagged, versioned scripts with deduplication by hash
- AI-backed analysis with scores, recommendations, and remediation notes
- Dual-mode discovery (keyword + vector search) plus documentation crawler
- Analytics and audit trails for compliance and release readiness

Management rollout details live in [.../MANAGEMENT-PLAYBOOK.md](#).

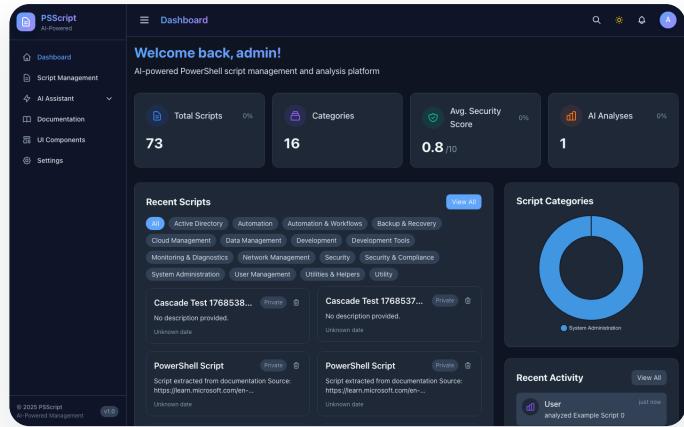
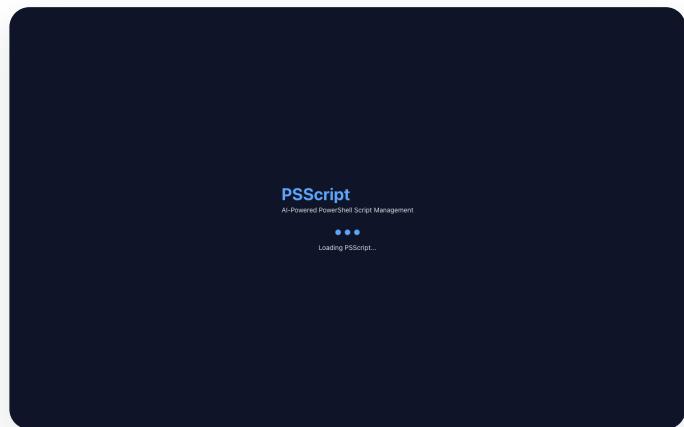
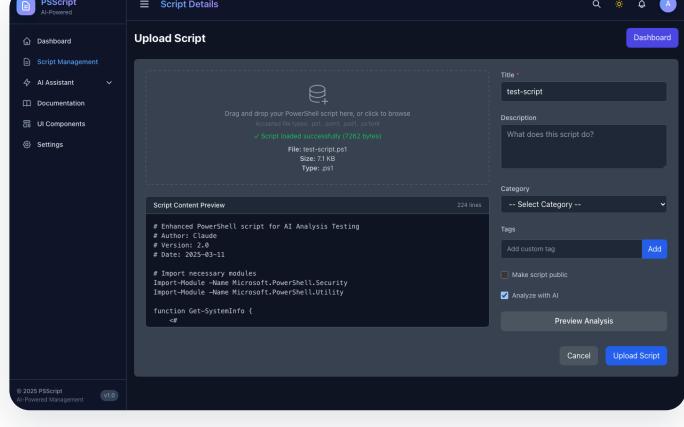
System overview

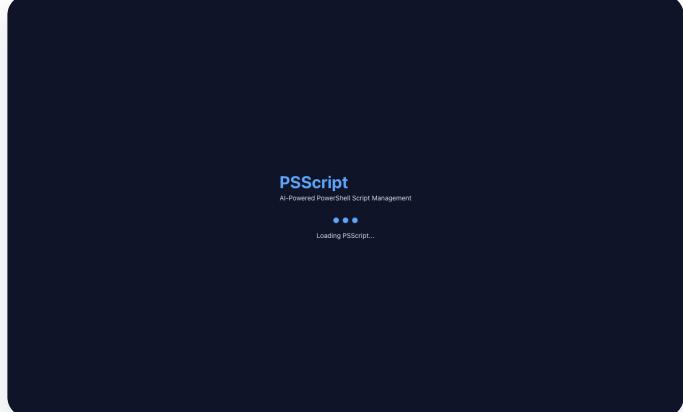
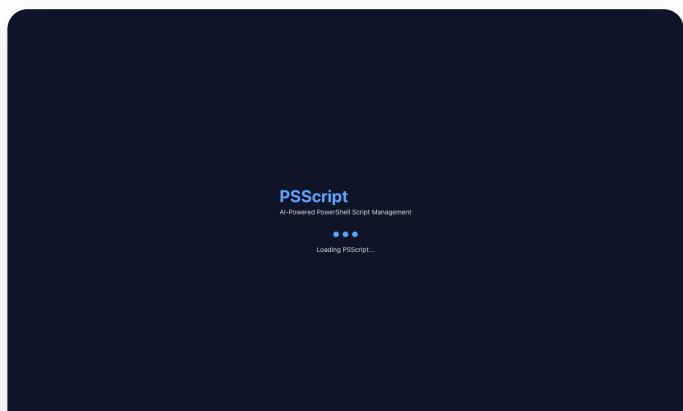


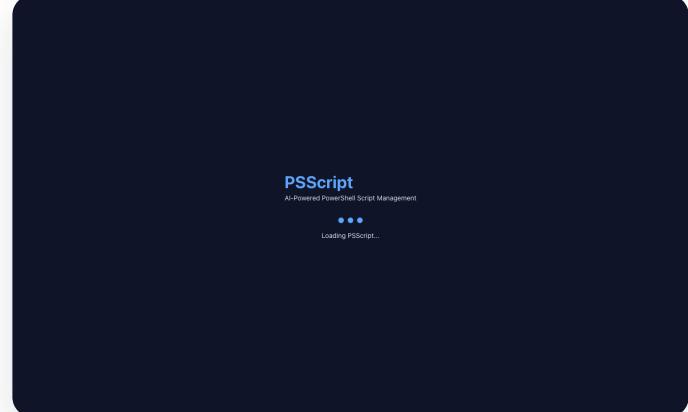
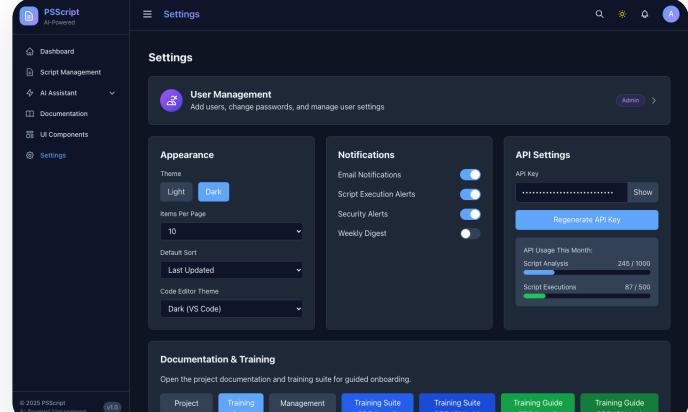
Sample Usage Metrics



Feature cheat sheet

Area	What to look for	Screenshot
Dashboard	Stats, recent scripts, security trends	
Script library	Categories, owners, version history	
Upload	Live preview, tags, dedup hints	

Area	What to look for	Screenshot
Analysis	Scores, findings, remediation	 <p>PSScript AI-Powered PowerShell Script Management ••• Loading PSScript...</p>
Docs + crawl	Cmdlet search, crawling, saved excerpts	 <p>PSScript AI-Powered PowerShell Script Management ••• Loading PSScript...</p>
Chat/agents	Quick prompts, agentic orchestration	 <p>PSScript AI-Powered PowerShell Script Management ••• Loading PSScript...</p>

Area	What to look for	Screenshot
Analytics	Adoption, usage, training exports	
Settings	API usage, notifications, training links	

Training roadmap



Environment setup

Recommended mock mode so every lab works offline:

```
./start-all-mock.sh
```

Ports: - Frontend: <http://localhost:3002> - Backend API: <http://localhost:4000/api> -

AI service: <http://localhost:8000>

Verify: - `curl http://localhost:4000/health` → 200 OK - Open

<http://localhost:3002> and click **Use Default Login** - Confirm dashboard counters populate with mock data

Guided walkthrough

Ship a vetted script in ~15 minutes:

1. **Sign in** → use Default Login on the login screen.
2. **Scan the dashboard** → note total scripts, AI analyses, and security score trend.
3. **Upload** → go to Upload, add `test-script.ps1`, tags, and category; note dedup hints.
4. **Review detail** → open the new script, confirm metadata, owner, and activity feed.
5. **Analyze** → open Analysis, read findings, and log remediation notes.
6. **Search** → find a similar script via keyword + vector search.
7. **Consult docs** → open Documentation, search for a related cmdlet, and save an excerpt.
8. **Chat** → ask the assistant “safest way to schedule this script weekly?” and record the answer.
9. **Analytics** → confirm the run shows up in usage metrics and training exports.
10. **Settings** → open Documentation & Training and verify the PDF/DOCX links load.

Role-based paths

Role	Focus	Primary modules
Script author	Upload, analyze, improve	01, 02, 03
Security reviewer	Risk review, approvals	03, 04, 05
Platform admin	Operations, reliability	01, 05

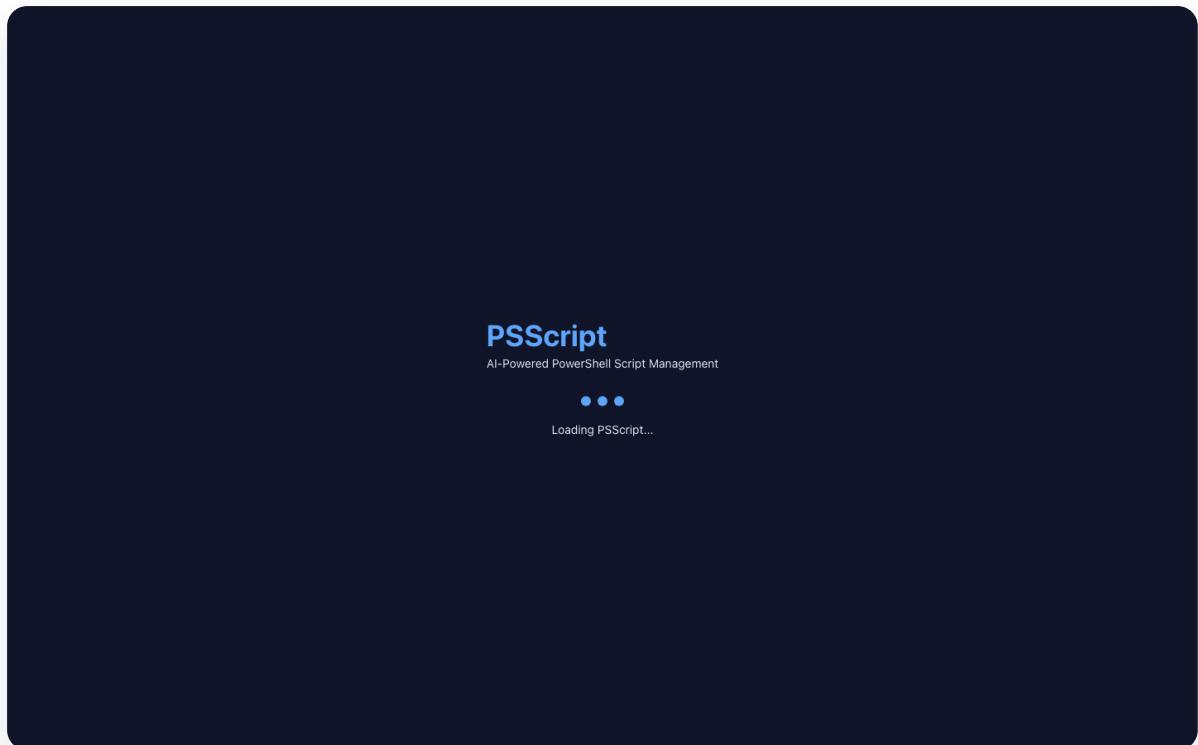
Module breakdown

Module	Objectives	Key screen
Module 01: Foundations	Navigate UI, find docs, understand services	Dashboard
Module 02: Script Lifecycle	Upload, tag, version scripts	Upload
Module 03: AI Analysis	Read scores and recommendations	Analysis
Module 04: Search	Use keyword + vector search	Scripts
Module 05: Operations	Review analytics and logs	Analytics

Hands-on labs

Lab 01: Sign in and orient

1. Open the login screen and use Default Login.
2. Review stats cards and activity feed on the dashboard.
3. Find Scripts, Documentation, Analytics in the sidebar.



Welcome back, admin!

AI-powered PowerShell script management and analysis platform

Total Scripts 73

Categories 16

Avg. Security Score 0.8 /10

AI Analyses 1

Recent Scripts

- 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

Script Categories

System Administration

Recent Activity

User analyzed Example Script 0 just now

Lab 02: Upload and analyze a script

1. Go to Upload and add `test-script.ps1` with tags + category.
2. Observe dedup hints; submit the script.
3. Open the script detail view and verify metadata.
4. Run AI analysis and capture recommendations.

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

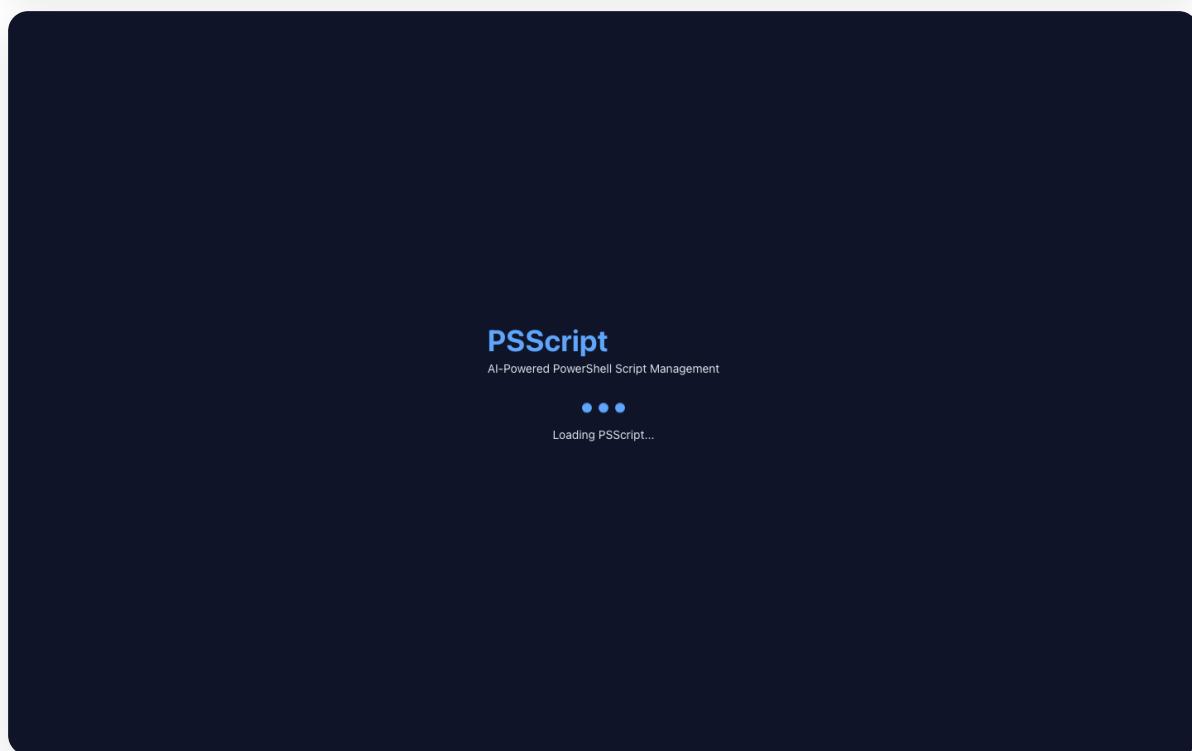
Make script public

Analyze with AI

Preview Analysis

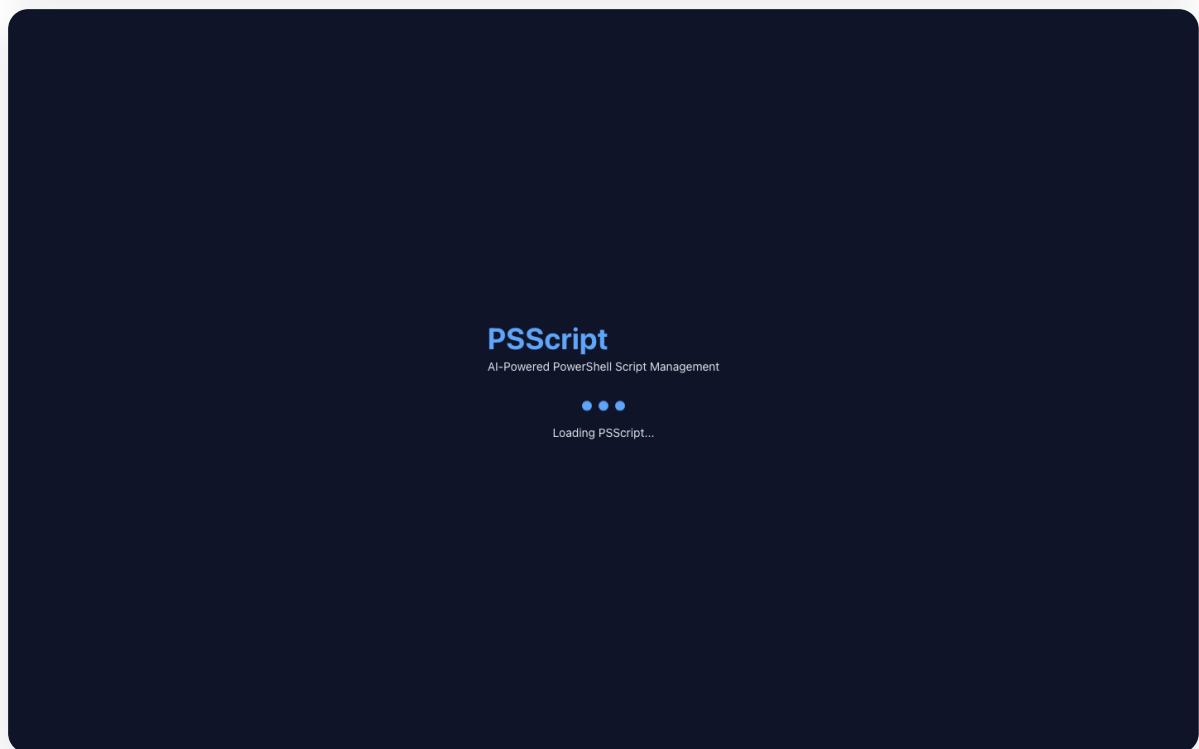
Cancel **Upload Script**

The screenshot shows the PSScript web application interface. On the left is a dark sidebar with navigation links: Dashboard, Script Management (selected), AI Assistant, Documentation, UI Components, and Settings. The main area has a title bar with 'Script Details' and a search icon. Below it is a 'Upload Script' section with a file upload input and a message: 'Drag and drop your PowerShell script here, or click to browse'. A green success message says 'Script loaded successfully (7262 bytes)'. Below that, details are shown: File: test-script.ps1, Size: 71 KB, Type: .ps1. A progress bar shows 'Uploading... 0%'. To the right, there are fields for 'Title' (test-script), 'Description' (What does this script do?), 'Category' (dropdown menu), 'Tags' (button to add custom tag, 'Add' button), 'Make script public' (checkbox), and 'Analyze with AI' (checkbox). A 'Preview Analysis' button is also present. At the bottom are 'Cancel' and 'Uploading...' buttons.



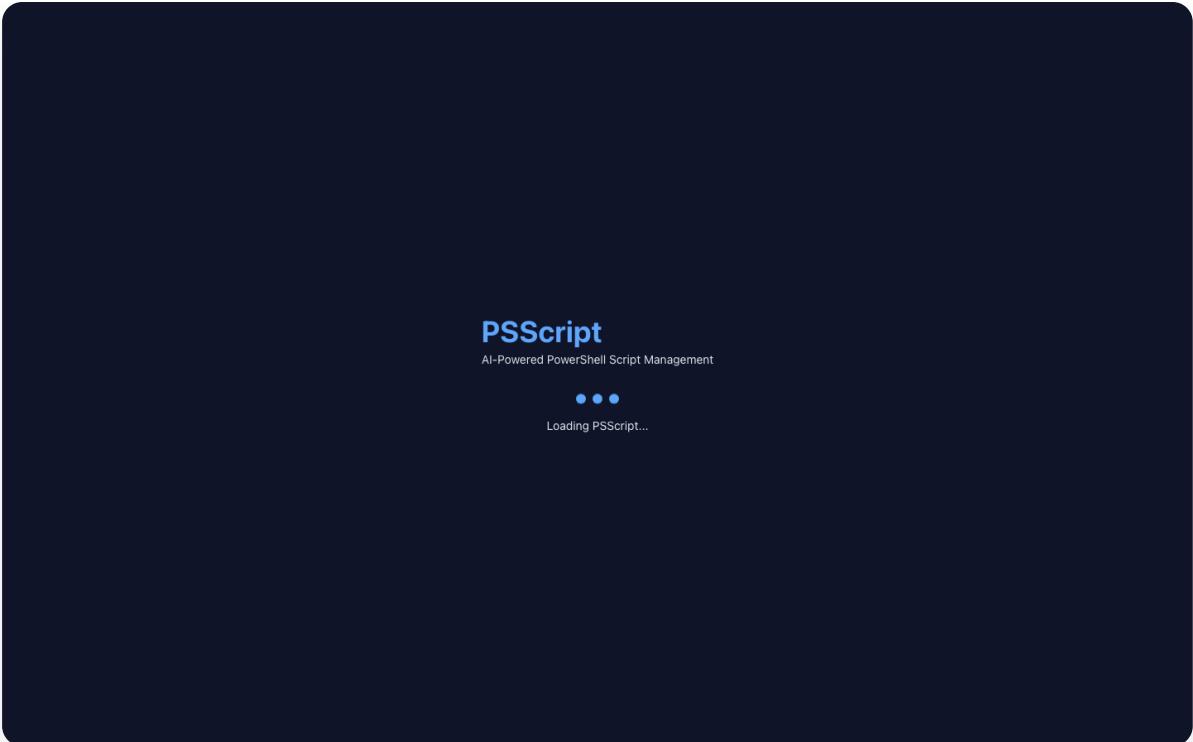
Lab 03: Documentation and AI chat

1. Open Documentation; search for a cmdlet (e.g., `Get-Process`).
2. Save an excerpt to your notes.
3. Open Chat and ask for safe usage patterns.



Lab 04: Analytics and governance

1. Open Analytics; review usage metrics and trend lines.
2. Identify top activity areas and any gaps.
3. Draft a governance checklist for releases.



Lab 05: Settings and training resources

1. Open Settings; review API usage and notifications.
2. Locate Documentation & Training and open the Training Suite.
3. Open the local PDF export at

<http://localhost:4000/docs/exports/pdf/Training-Guide.pdf>.

PSScript
AI-Powered

- Dashboard
- Script Management
- AI Assistant
- Documentation
- UI Components
- Settings

≡ Settings

User Management

Add users, change passwords, and manage user settings

Appearance

Theme: Dark (selected)

Items Per Page: 10

Default Sort: Last Updated

Code Editor Theme: Dark (VS Code)

Notifications

Email Notifications: On

Script Execution Alerts: On

Security Alerts: On

Weekly Digest: Off

API Settings

API Key: (Show)

Regenerate API Key

API Usage This Month:

Category	Usage	Quota
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)

Sample actions

Upload via API

```
curl -X POST http://localhost:4000/api/scripts \  
-H "Authorization: Bearer <token>" \  
-F "file=@test-script.ps1" \  
-F "title=Reset-UserPassword" \  
-F "description=Reset AD user password" \  
-F "tags=security,active-directory"
```

Run analysis

```
curl -X POST http://localhost:8000/analyze \  
-H "Content-Type: application/json" \  
-d '{"content": "Get-ADUser -Filter *", "type": "security"}'
```

Vector search

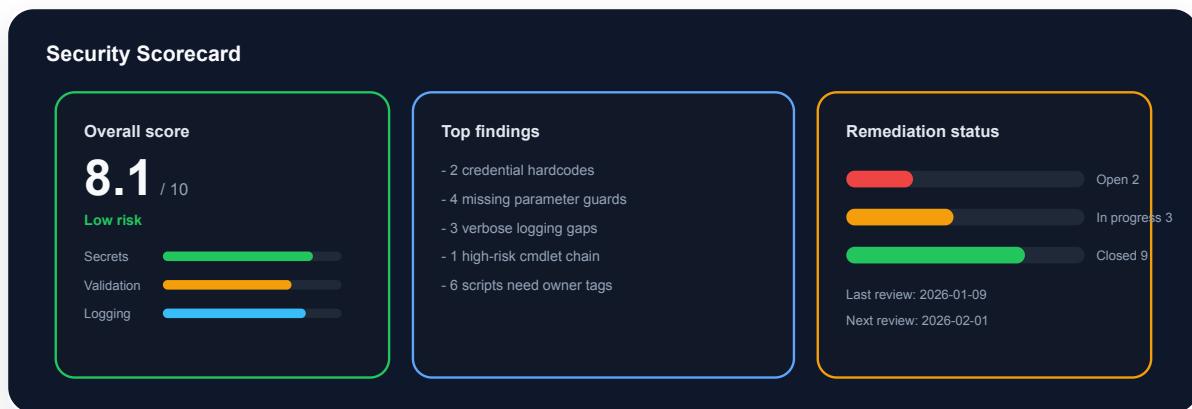
```
curl -X POST http://localhost:8000/similar \  
-H "Content-Type: application/json" \  
-d '{"content": "active directory onboarding", "limit": 5}'
```

Operational readiness

Governance checklist

- Scripts are tagged, categorized, and have an owner
- Security score is reviewed and exceptions are documented
- Duplicates are prevented by hash check
- Analysis results are logged and stored
- Analytics reflect script usage and review cadence

Security scorecard



Scorecard rubric

Signal	Description	Target
Security score	Weighted risk score from AI analysis	≥ 7.5
High-risk findings	Count of high severity issues	0
Remediation SLA	Days to close high-risk findings	≤ 7
Ownership coverage	Scripts with owner tag	100%
Review cadence	Days between audits	≤ 30

Operational checks

Area	Check	Target
Backend	Health endpoint	200 OK
Database	pgvector enabled	true
AI service	Analyze endpoint	200 OK
Cache	Redis reachable	Optional
UI	Dashboard load	< 3 seconds

Appendix

- Troubleshooting and support: `../SUPPORT.md`
- Screenshot capture: `./scripts/capture-readme-screenshots.sh`
- Exports (HTML/PDF/DOCX): `scripts/export-docs.sh --all`

Appendix: Troubleshooting

Issue	Likely cause	Fix
Upload fails	Backend not running	Start backend on port 4000
Analysis empty	AI service down	Start AI service or mock mode
Search returns none	Embeddings missing	Run embedding generation
Docs empty	Crawl not completed	Run documentation crawl

Generated 2026-01-16 21:23 UTC