

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](#)

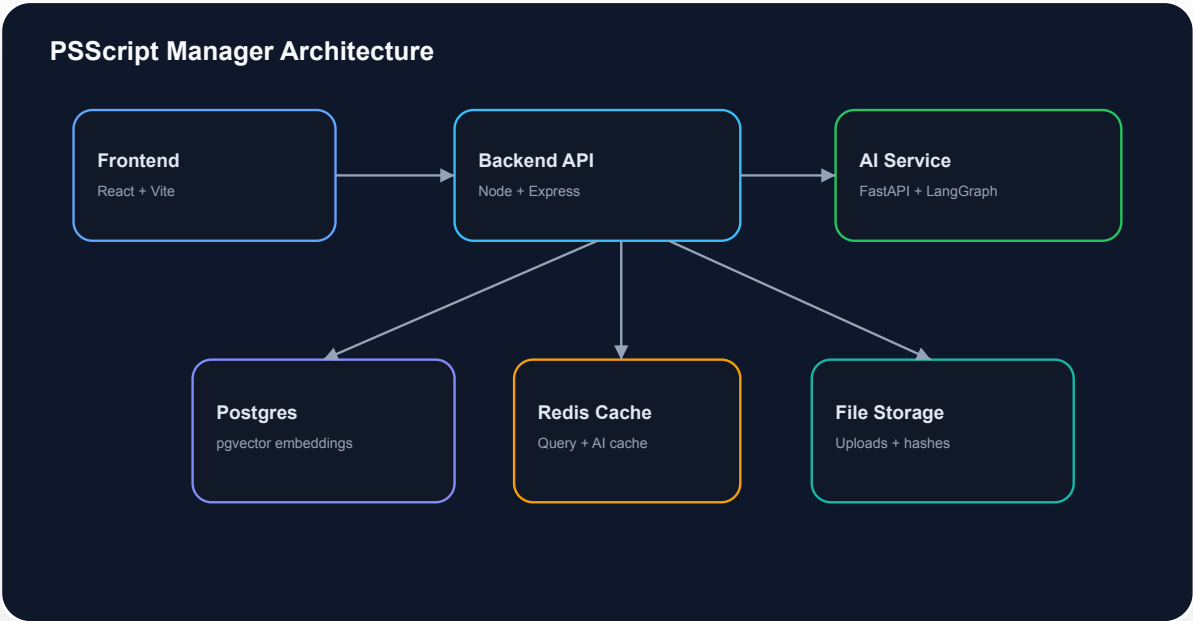
Program overview

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

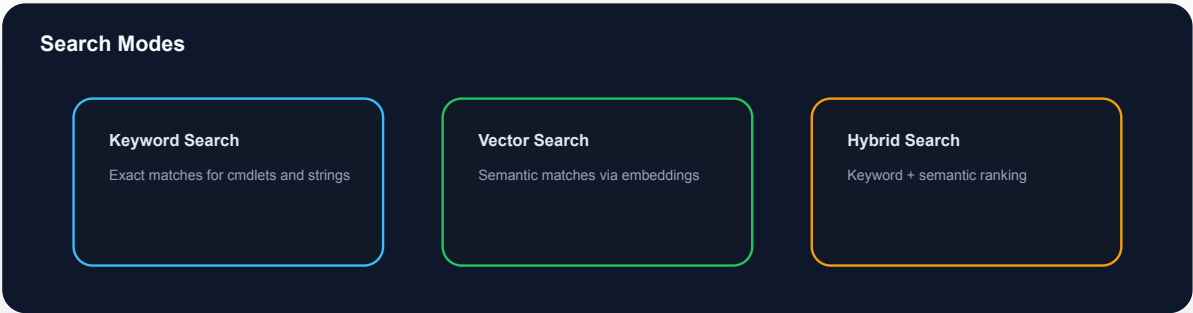
- Organized, tagged, versioned scripts with deduplication by hash
- AI-backed analysis with scores, recommendations, and remediation notes
- Dual-mode discovery (keyword + vector search) plus documentation crawler
- Analytics and audit trails for compliance and release readiness

Management rollout details live in `./MANAGEMENT-PLAYBOOK.md`.

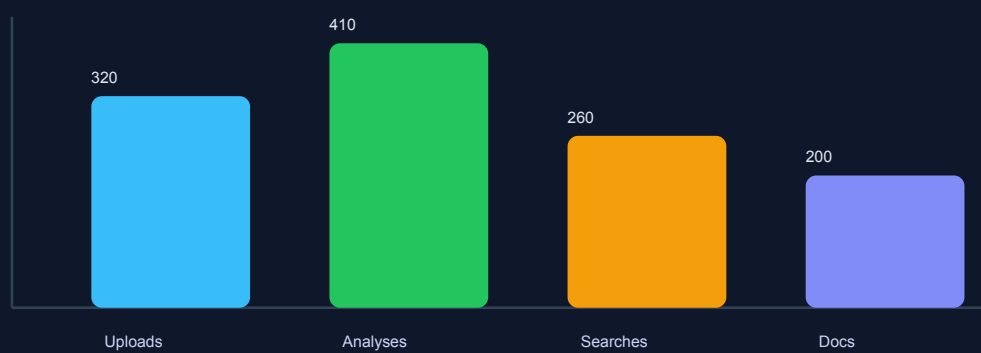
System overview



Analysis Pipeline

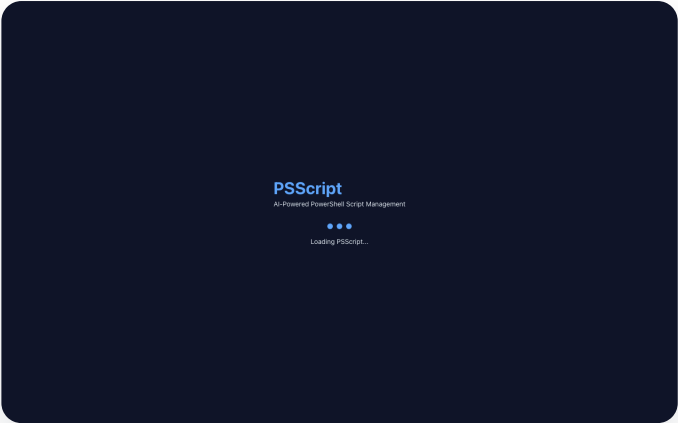
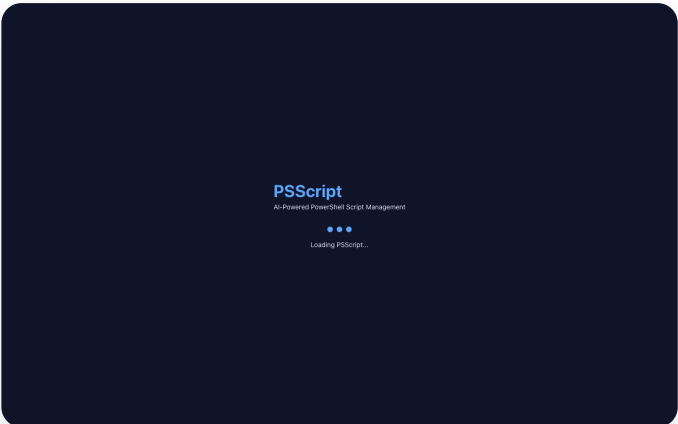
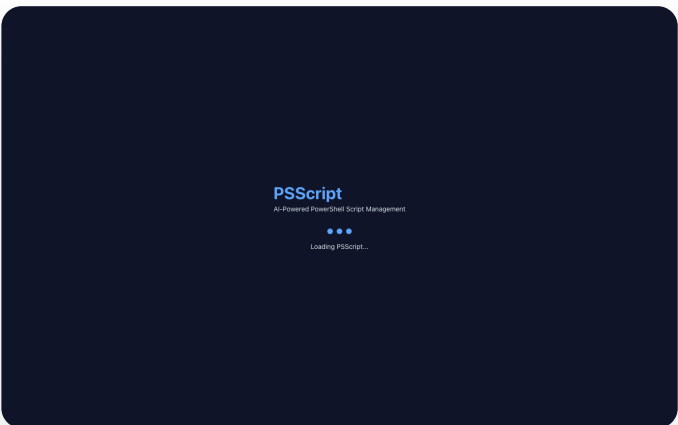


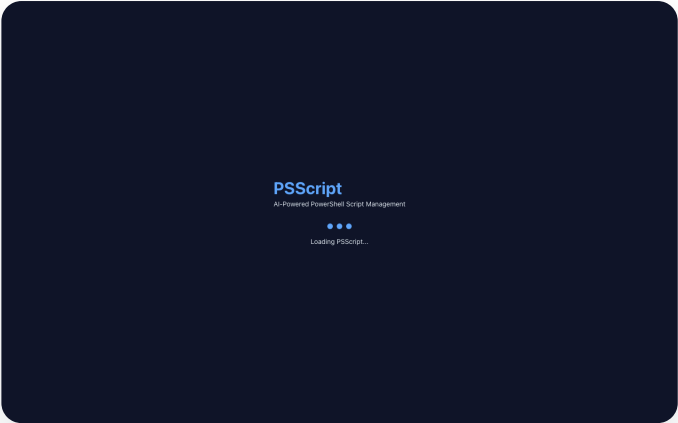
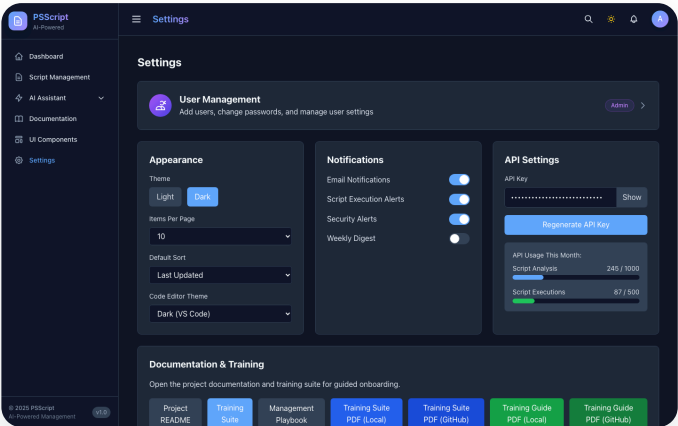
Sample Usage Metrics



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	
Docs + crawl	Cmdlet search, crawling, saved excerpts	
Chat/agents	Quick prompts, agentic orchestration	

Area	What to look for	Screenshot
Analytics	Adoption, usage, training exports	 A dark-themed loading screen for PSScript. The text 'PSScript' is displayed in blue, with the subtitle 'AI-Powered PowerShell Script Management' below it. Three small blue dots are arranged horizontally, and the text 'Loading PSScript...' is at the bottom.
Settings	API usage, notifications, training links	 A screenshot of the PSScript Settings page. The left sidebar contains links: Dashboard, Script Management, AI Assistant, Documentation, UI Components, and Settings. The main content area is titled 'Settings' and includes sections for 'User Management' (with an 'Admin' link), 'Appearance' (with options for Theme, Items Per Page, Default Sort, and Code Editor Theme), 'Notifications' (with toggle switches for Email Notifications, Script Execution Alerts, Security Alerts, and Weekly Digest), and 'API Settings' (with an API Key field, a 'Show' button, a 'Regenerate API Key' button, and usage statistics for 'API Usage This Month'). At the bottom, there is a 'Documentation & Training' section with links to Project README, Training Suite, Management Playbook, Training Suite PDF (Local), Training Suite PDF (GitHub), Training Guide PDF (Local), and Training Guide PDF (GitHub).

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

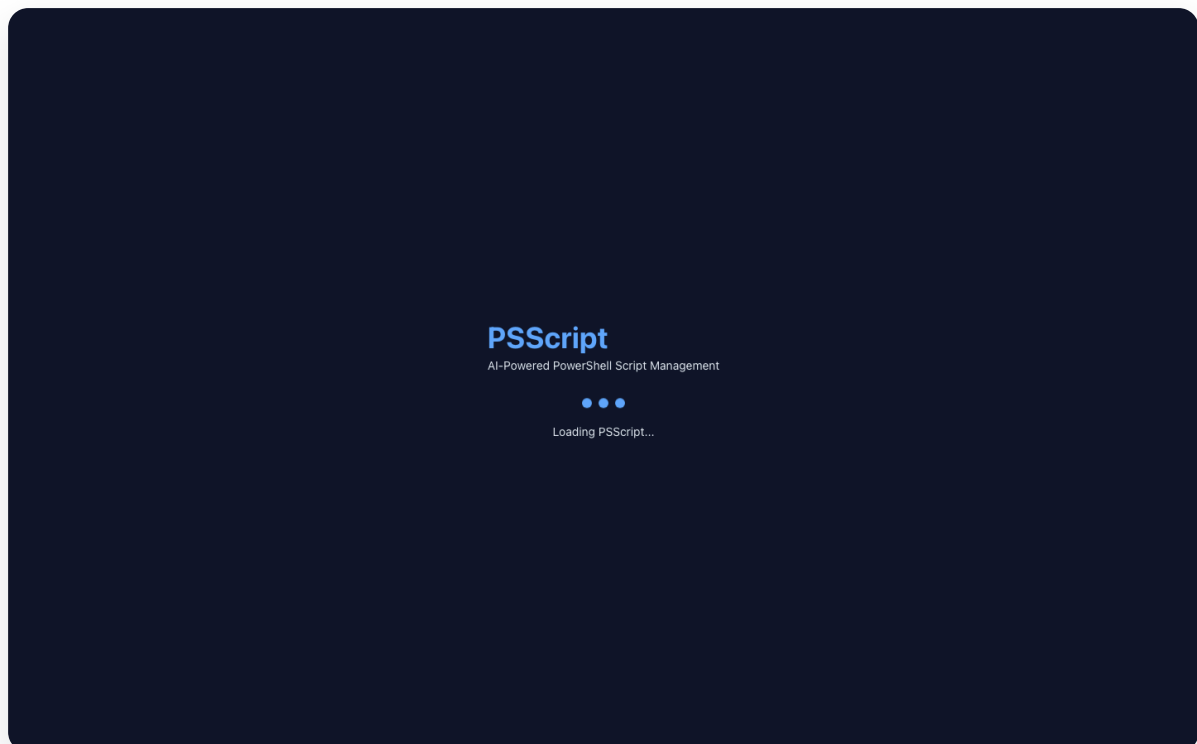
Module breakdown

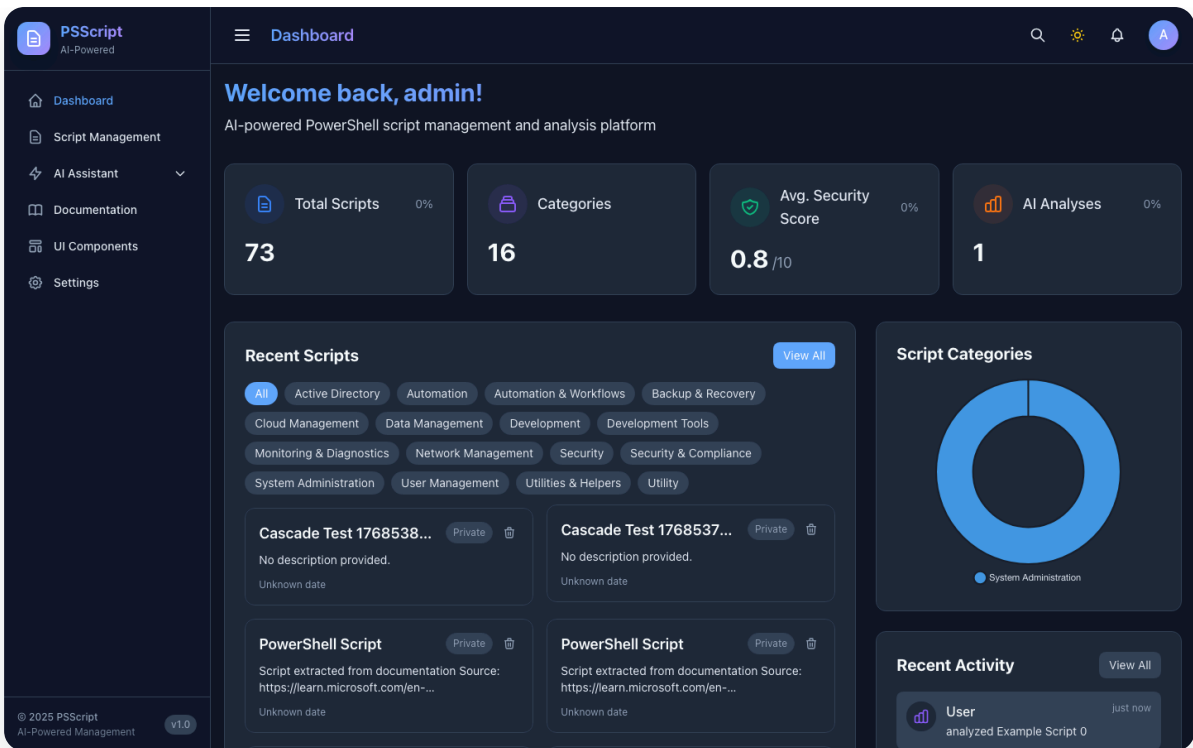
Module	Objectives	Key screen
Module 01: Foundations	Navigate UI, find docs, understand services	Dashboard
Module 02: Script Lifecycle	Upload, tag, version scripts	Upload
Module 03: AI Analysis	Read scores and recommendations	Analysis
Module 04: Search	Use keyword + vector search	Scripts
Module 05: Operations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

Hands-on labs

Lab 01: Sign in and orient

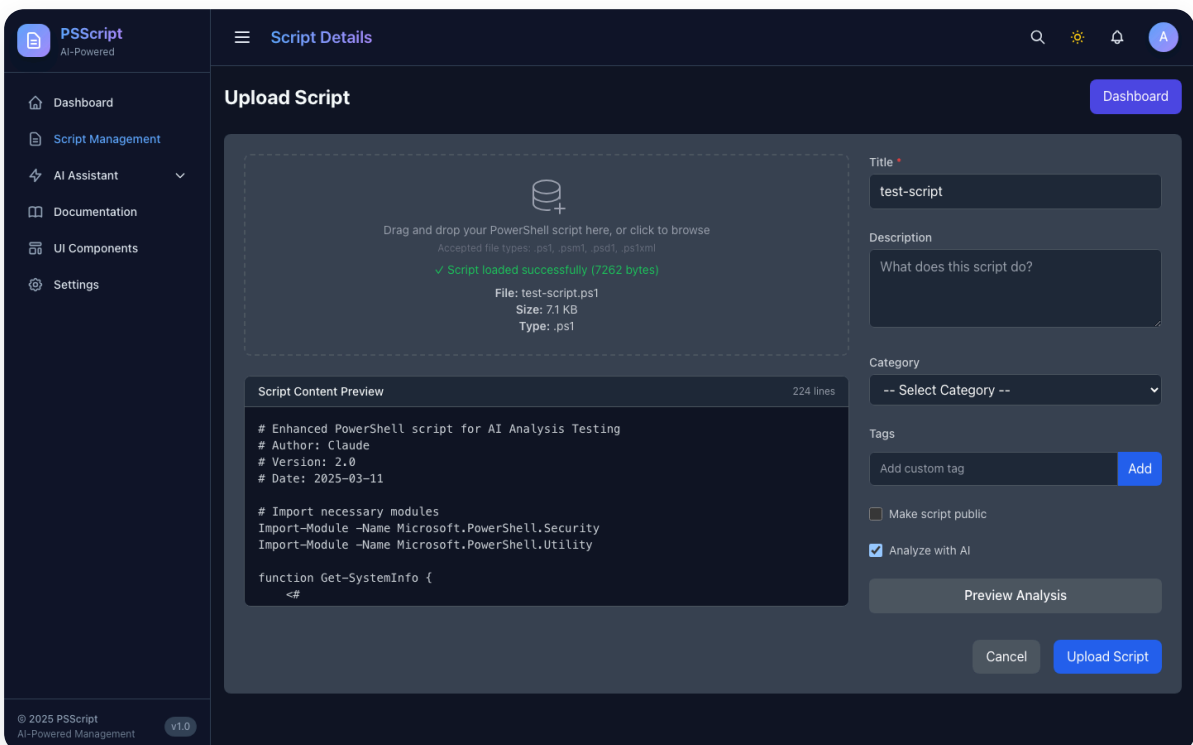
1. Open the login screen and use Default Login.
2. Review stats cards and activity feed on the dashboard.
3. Find Scripts, Documentation, Analytics in the sidebar.

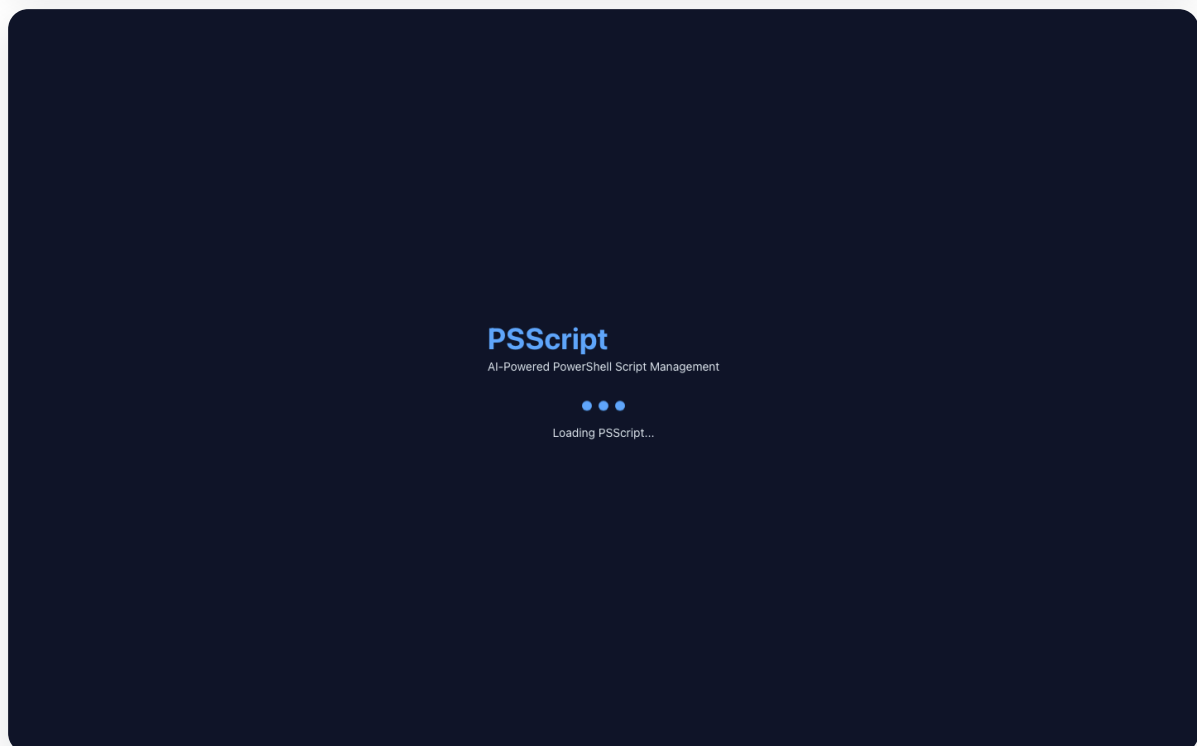
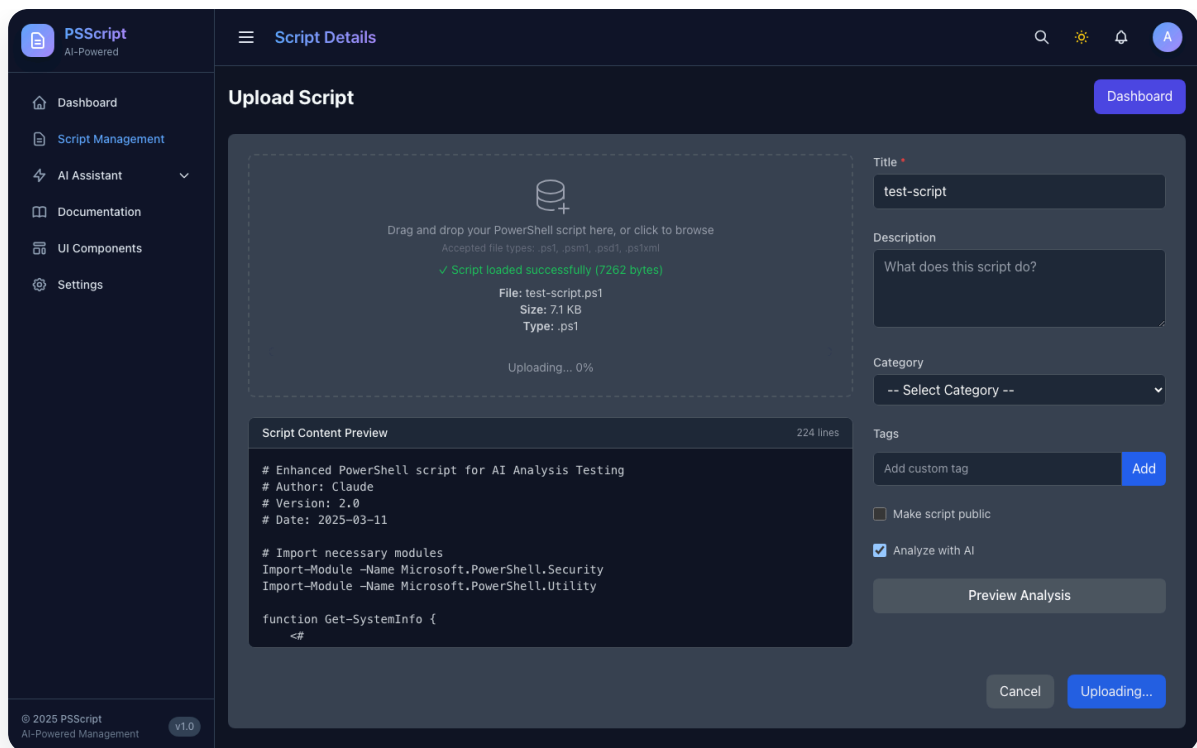




Lab 02: Upload and analyze a script

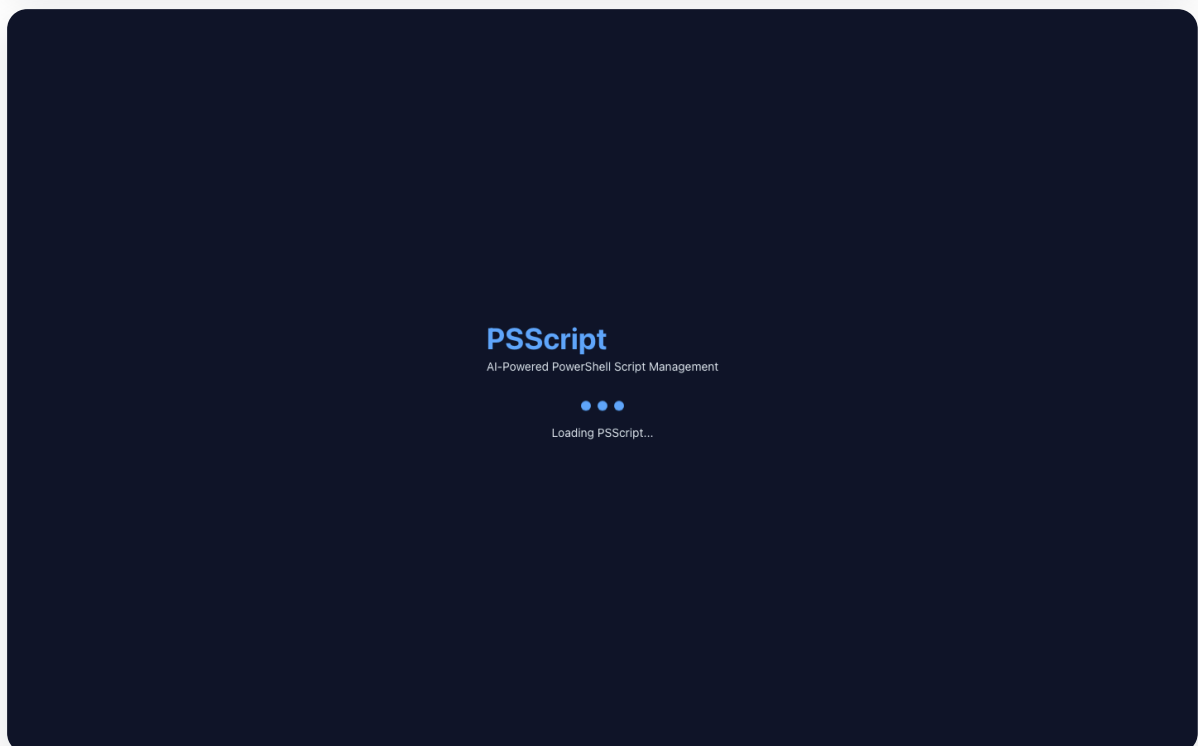
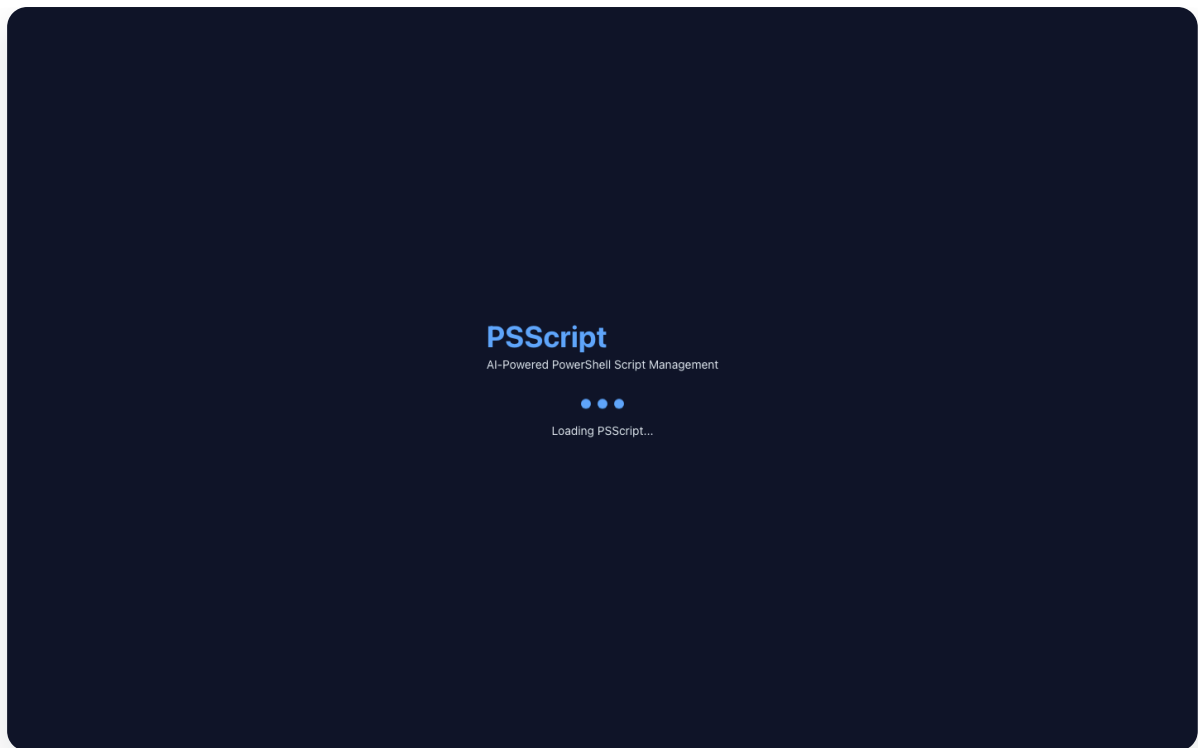
1. Go to Upload and add `test-script.ps1` with tags + category.
2. Observe dedup hints; submit the script.
3. Open the script detail view and verify metadata.
4. Run AI analysis and capture recommendations.





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.

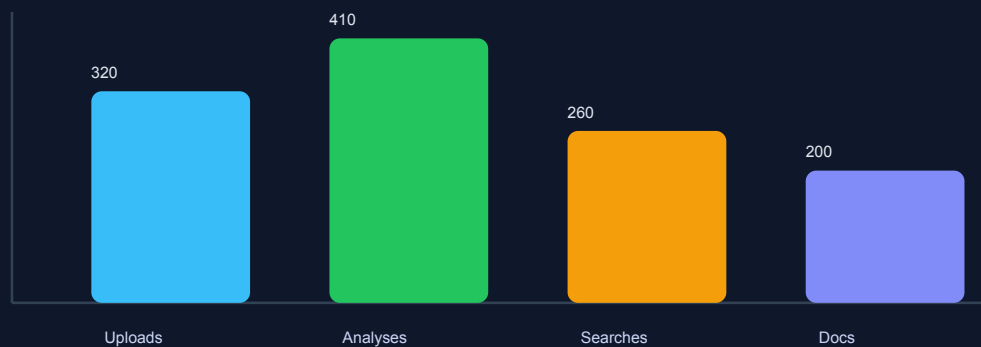
PSScript

AI-Powered PowerShell Script Management



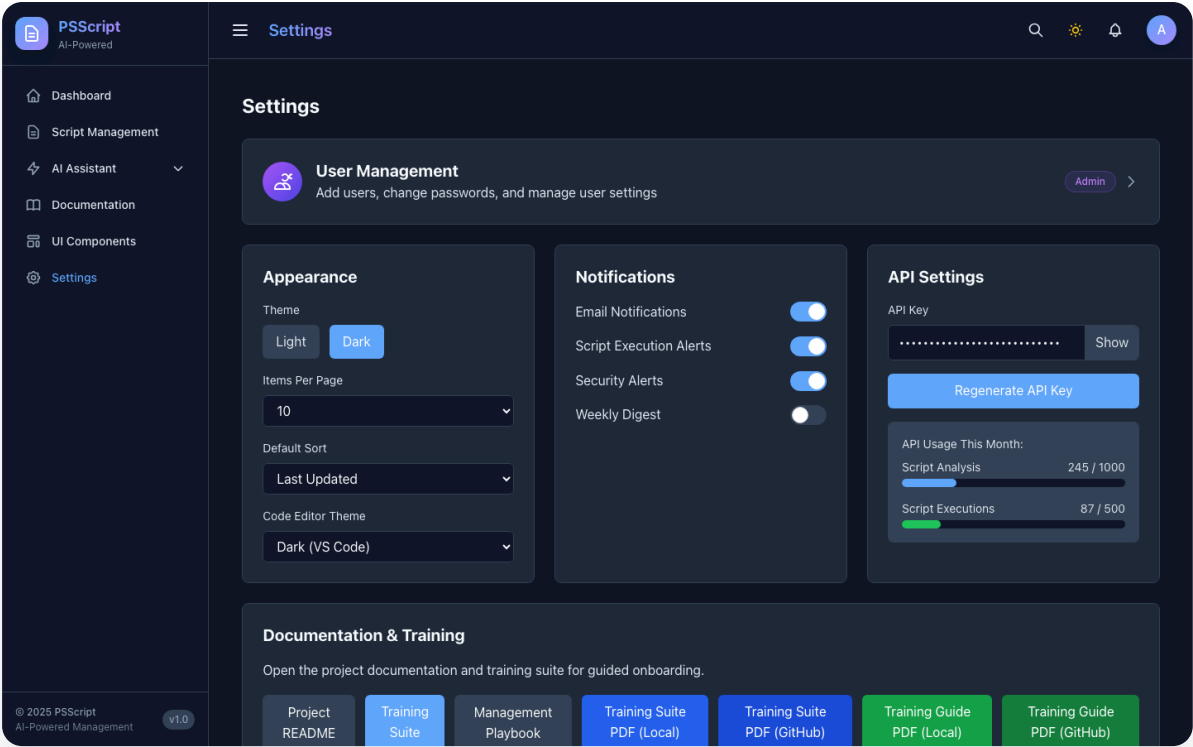
Loading PSScript...

Sample Usage Metrics



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`.



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