University of Bahrain
College of Information Technology
Department of Information Systems
B.Sc. In Cybersecurity

SENIOR PROJECT

ACADEMIC YEAR 2025-2026-SEMESTER 2

SECURE NATURAL LANGUAGE TO BASH EXECUTION - SHELLSENTRY

19 May 2026

Prepared by:

Aljazi Ali Almujaddam - 202202227

Dana Hussain Alhayki - 202203431

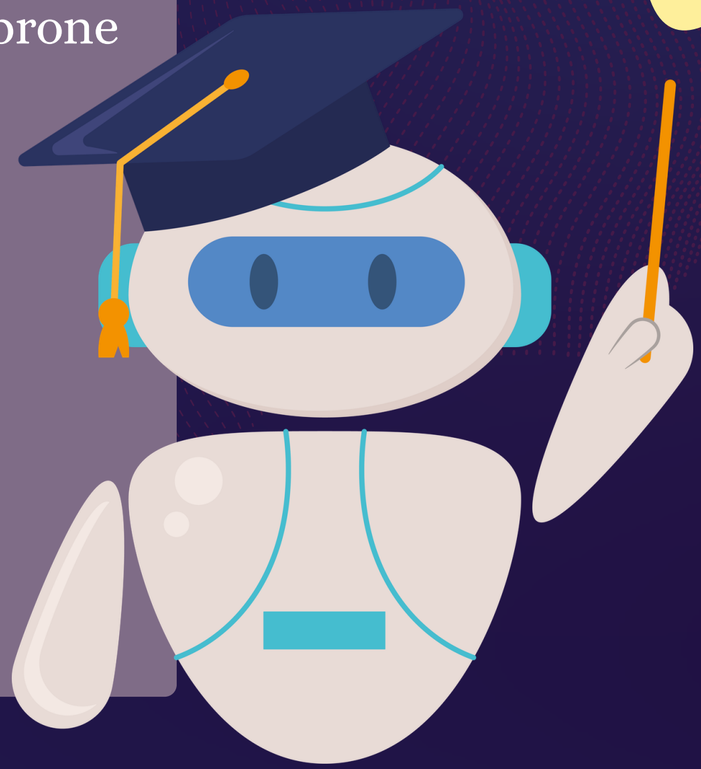
Supervised By :

Dr. Abdulla Khalifa Aldoseri

PROBLEM STATEMENT

- Linux administration requires Bash expertise
- Manual execution across multiple servers is repetitive and error-prone
- Human mistakes can cause outages or security incidents
- LLMs can generate Bash, but outputs may be unsafe:
- Hallucinated destructive commands
- Prompt injection
- Privilege abuse
- Direct remote execution risks

Need for a secure AI-assisted execution platform



OBJECTIVES & CONTRIBUTIONS

- Secure NL → Bash execution
- Multi-server SSH orchestration
- Deterministic command validation
- Host-aware command generation
- RAG grounding with trusted examples
- Audit logging and reporting
- Script archive + Safe scheduling

A major contribution is shifting from “AI assistant” toward a secure operational execution platform.

RELATED WORK SUMMARY

Study	Focus	Limitation
ScriptSmith	Bash generation + refinement	No secure execution
NL→Bash LLM Study	Functional correctness evaluation	No deployment platform
LLM-as-a-Judge	Validation/refinement	Relies on LLM judgment
Shell-GPT	Real-world command assistant	No execution safety enforcement

Research gap: Secure operational execution

RESEARCH GAP

Existing solutions provide:

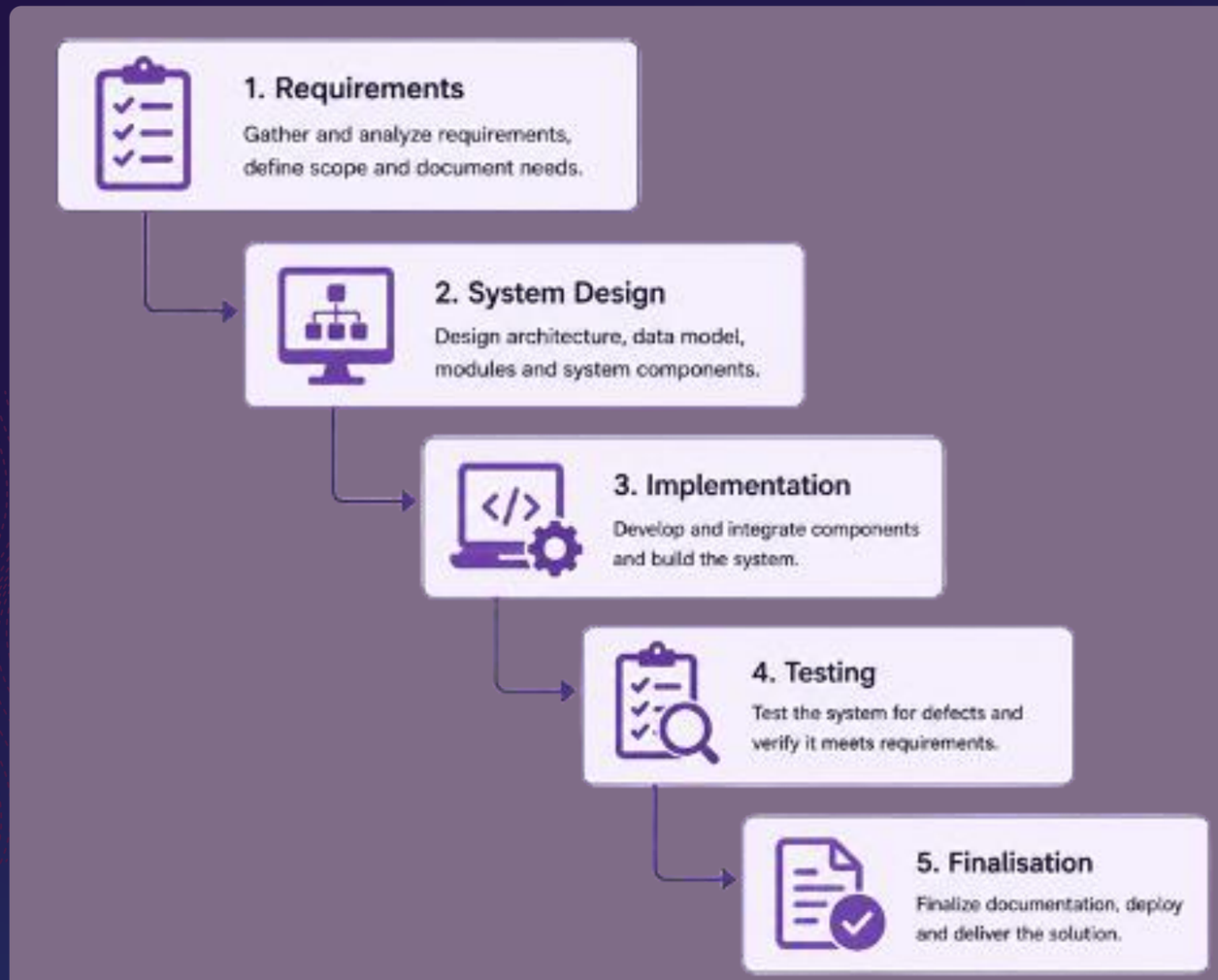
- AI assistance
- Bash generation
- Command refinement

But lack:

- Audit logging
- Safe cron automation
- Deterministic validation
- Authenticated execution
- SSH multi-host orchestration

Then: ShellSentry addresses these gaps

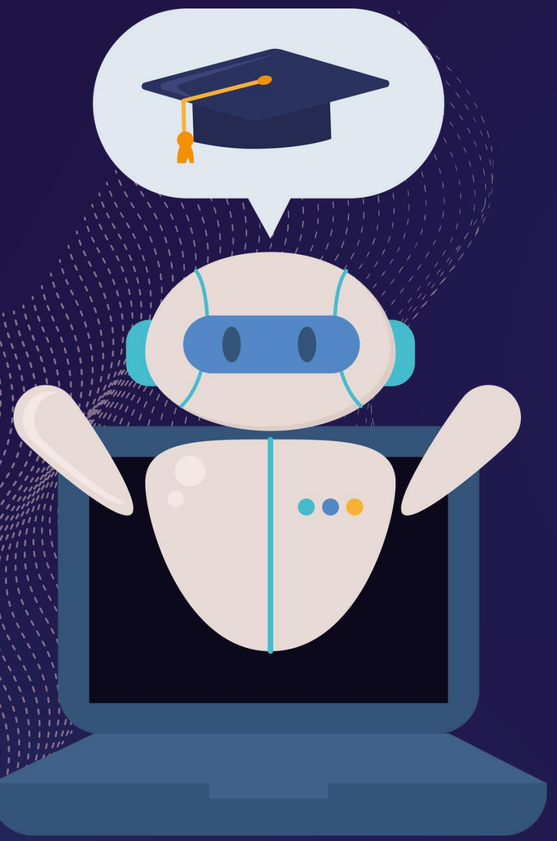
DEVELOPMENT METHODOLOGY



Reason:

Structured academic development

REQUIREMENT COLLECTION



Interviews with 2 cybersecurity specialist

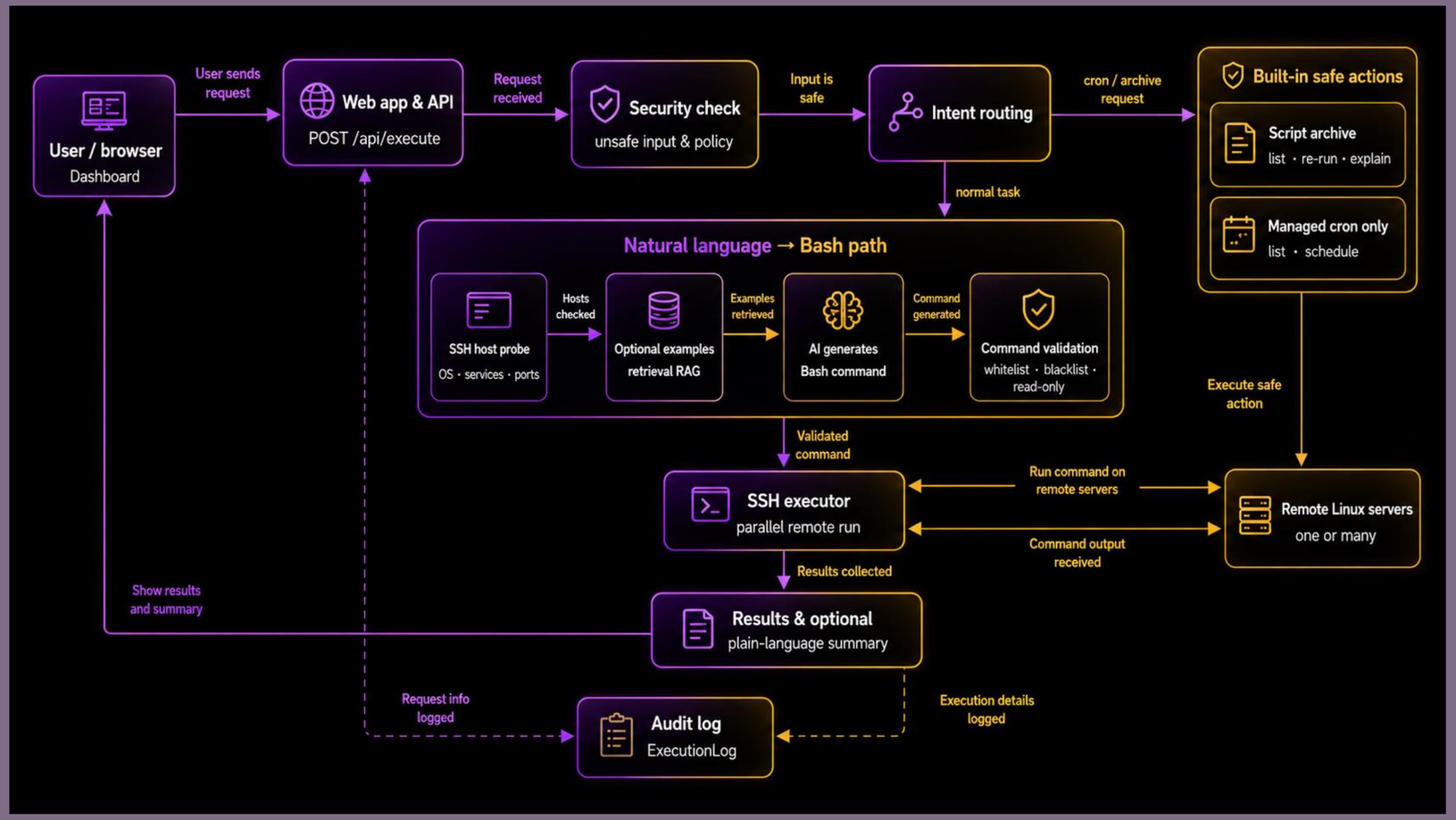
Key findings:

- Read-only by default
- Validation is mandatory
- Auditability required
- Least privilege
- Unsafe LLM output cannot be trusted

FUNCTIONAL REQUIREMENTS

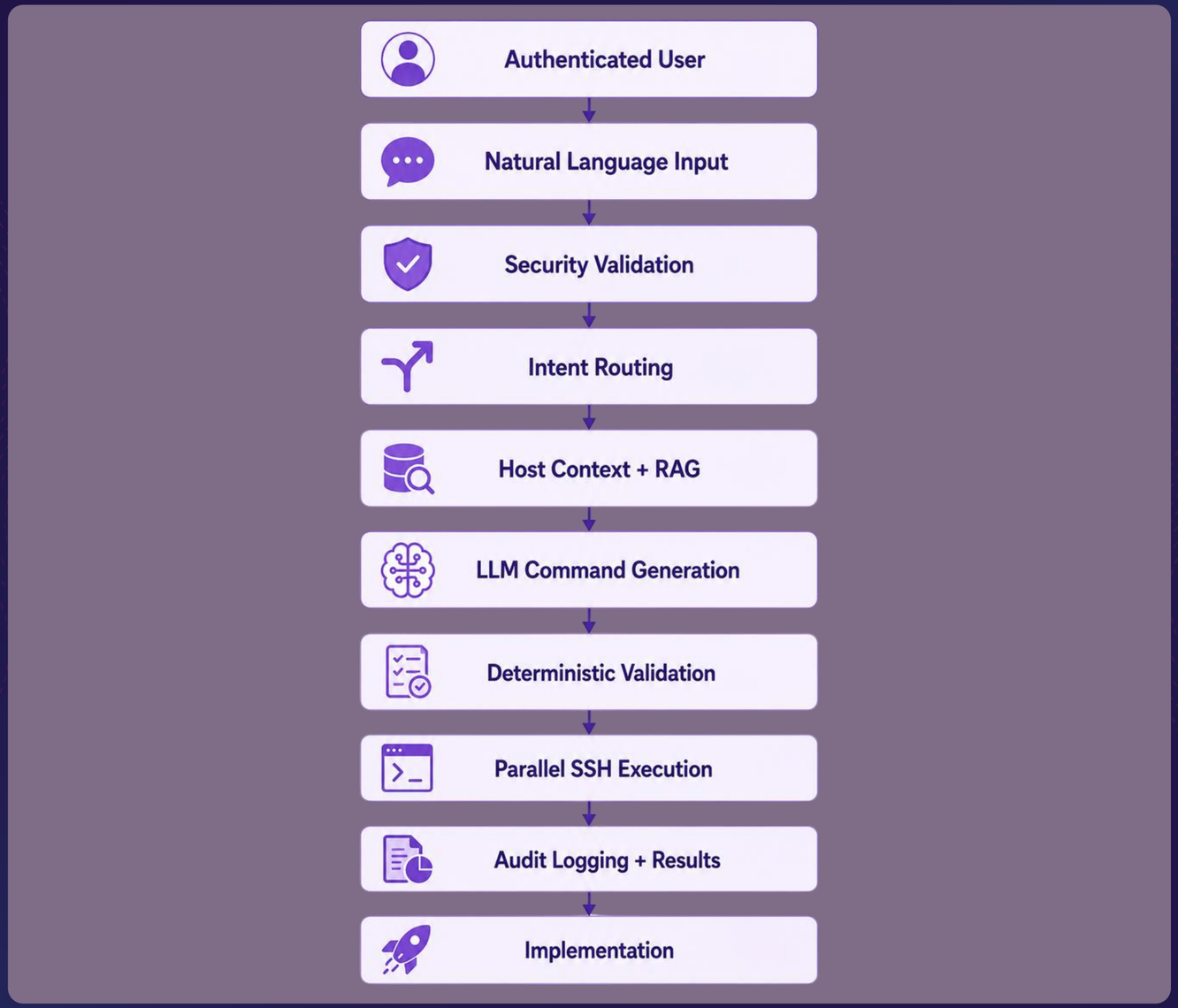
- NL → Bash generation
- Host-aware context collection
- RAG retrieval
- Command validation
- SSH authentication
- Multi-server execution
- Script archive
- Safe scheduling via Cron
- Audit logging
- Configurable security policies

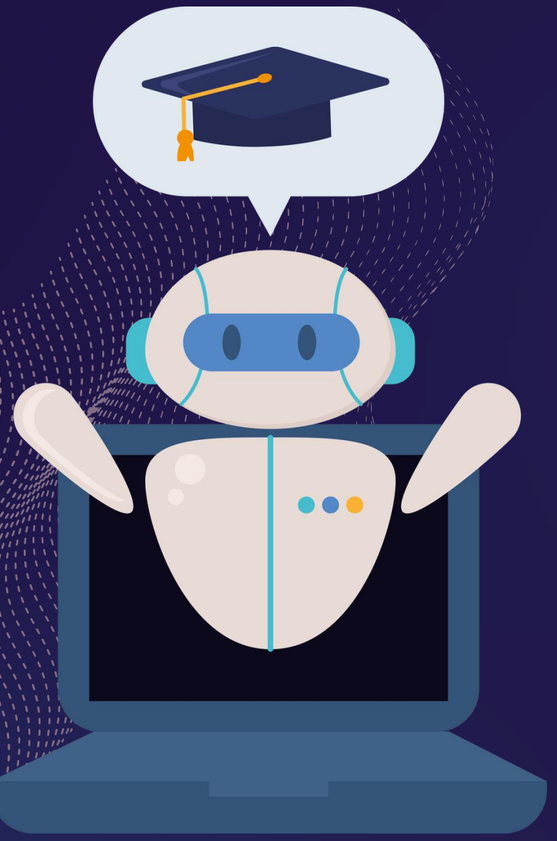
SYSTEM ARCHITECTURE



Explain flow: User → Flask → Security → Routing → RAG → LLM → Validator → SSH → Results → Database

EXECUTION PROTOCOL





SYSTEM DEMONSTRATION

IMPLEMENTATION STACK

Component	Technology
Frontend	Jinja/HTML/CSS/JS
Backend	Flask
Authentication	Flask-Login
Database	SQLite
SSH	Paramiko
AI model	llama-3.1-8b-instant / llama-3.3-70b versatile
RAG	Sentence Transformers + FAISS

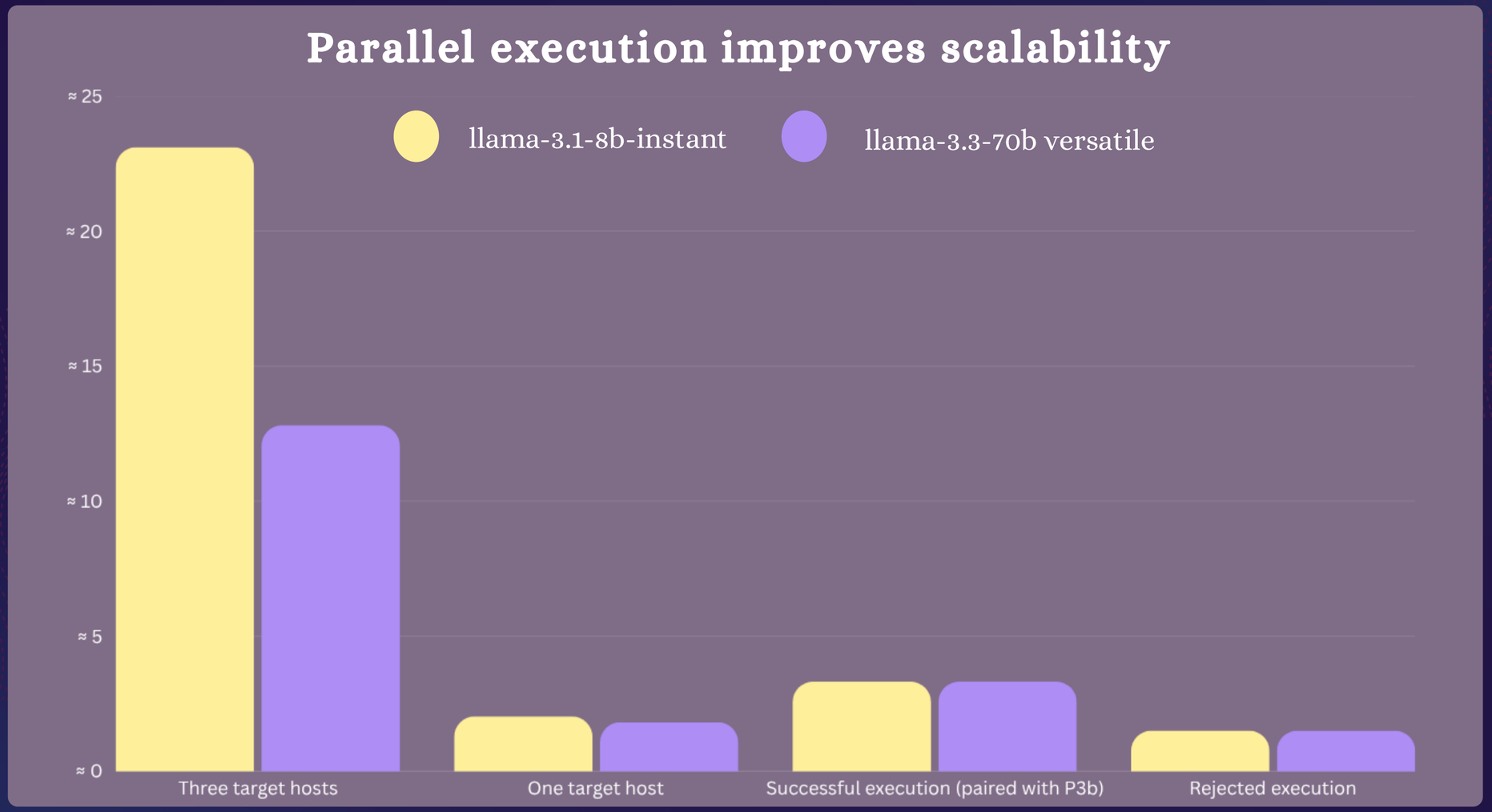
FUNCTIONAL TESTING RESULTS

Test	Result
Authentication	Pass
NL→Bash	Pass
Multi-host execution	Pass
Archive scripts	Pass
Safe scheduling via Cron	Pass

SECURITY TESTING RESULTS

Security Test	Result
SQL Injection	Pass
Prompt Injection	Pass
Unsafe Command	Blocked
Root Execution	Blocked
Cron Deletion	Blocked
SSH Credential Failure	Handled

PERFORMANCE RESULTS



COMPARISON WITH Shell-GPT

ShellSentry focuses on secure execution, not just generation

Feature	ShellSentry	Shell-GPT
Bash generation	Yes	Yes
Validation	Yes	Limited
SSH execution	Yes	No
Multi-host	Yes	No
Audit logging	Yes	No
Safe scheduling via Cron	Yes	No

CONCLUSION

- Secure AI-assisted Linux administration prototype
- Validated NL→Bash execution
- Multi-host SSH orchestration
- Strong security enforcement
- Successful testing results

FUTURE Work

- RBAC
- Human approval workflow
- Dry-run mode
- SIEM integration
- Production deployment

THANK YOU FOR YOUR ATTENTION

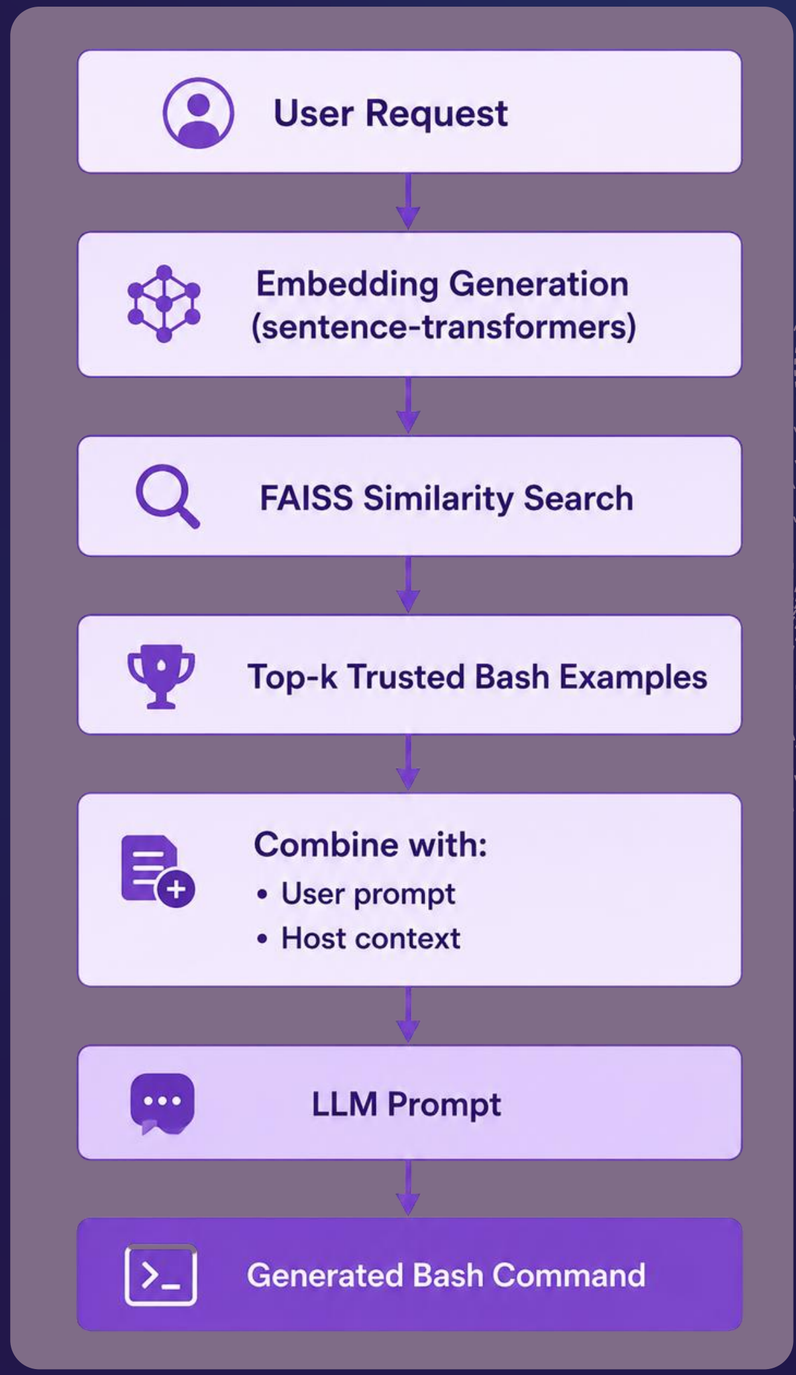
We sincerely appreciate your time, interest, and support throughout our presentation.

(وَآخِرُ دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ)

اللهم لك الحمد حتى ترضى ولك الحمد إذا رضيت ولك الحمد بعد الرضى.

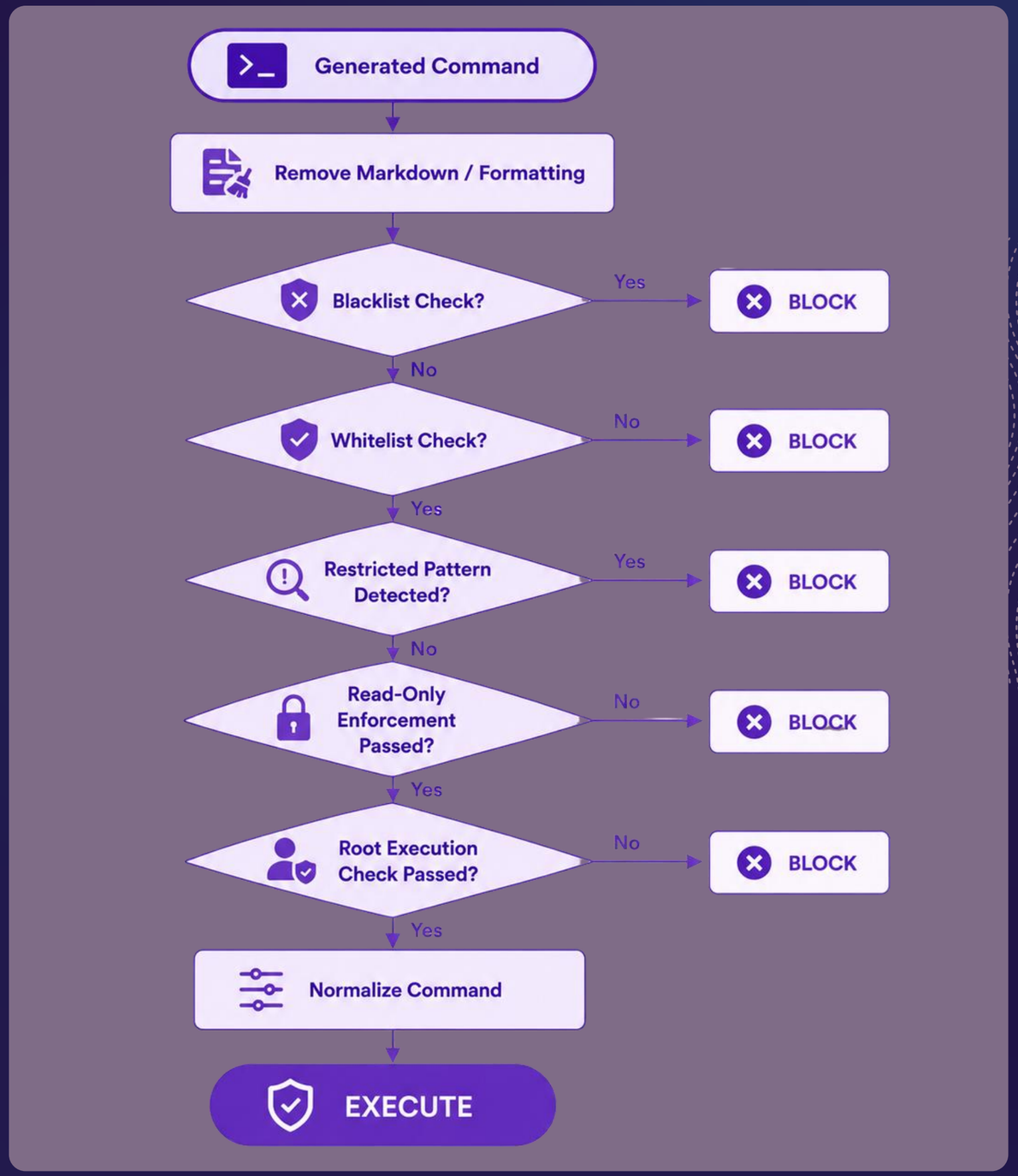
ANY QEUSTIONS?

RETRIEVAL AUGMENTED GENERATION PIPELINE



Extras

DETERMINISTIC COMMAND VALIDATION



Extras

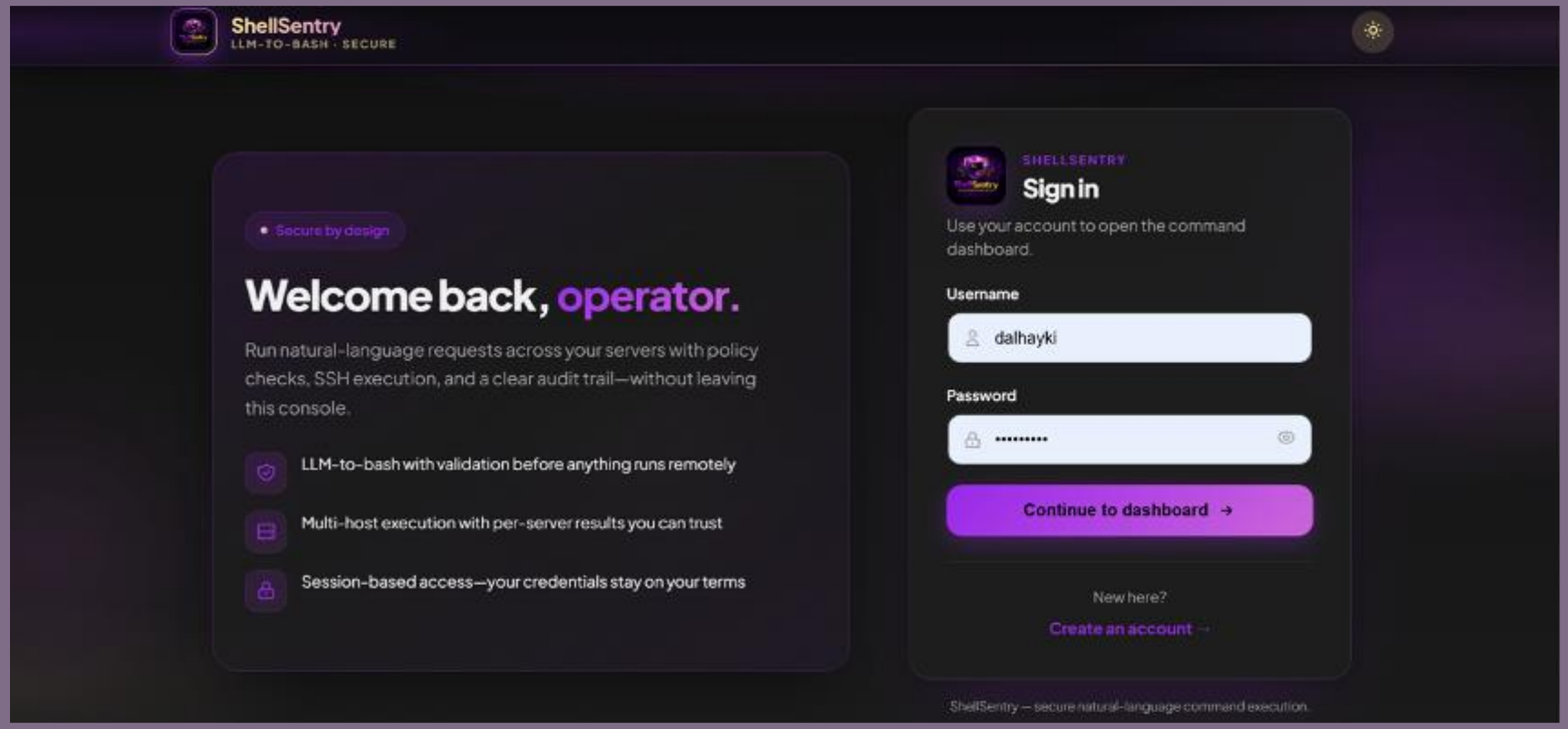
SYSTEM CONFIGURATION CONTROIs

- environment variables
- no hardcoded secrets
- least privilege

Setting	Default
READ_ONLY_EXECUTION	TRUE
ALLOW_ROOT_EXECUTION	FALSE
SSH authentication	Password / Key / Agent
REMOTE_SERVERS	Configurable
SERVER_CREDENTIALS	Per-host supported
LLM provider	Configurable

Extras

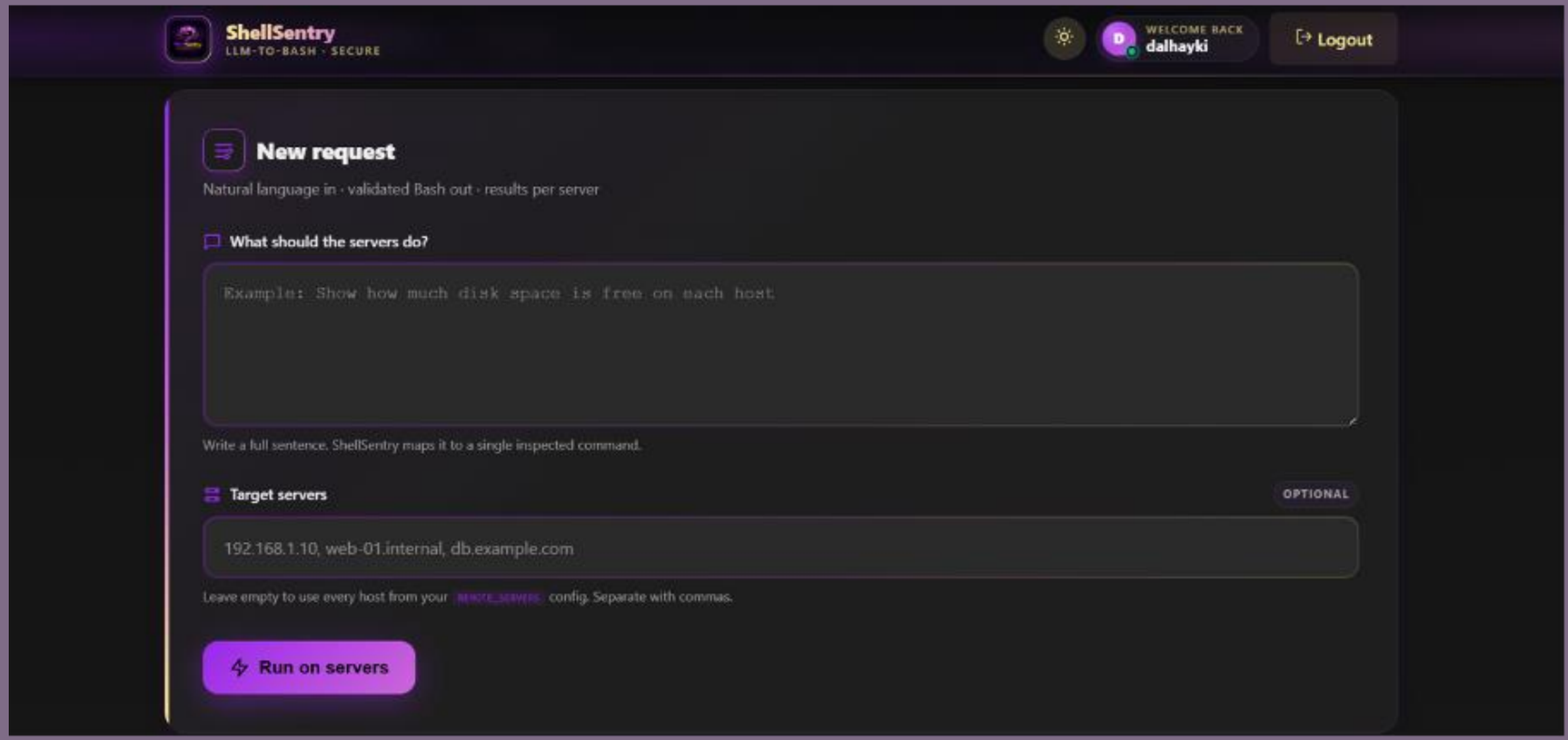
SYSTEM INTERFACE



Login page

Extras

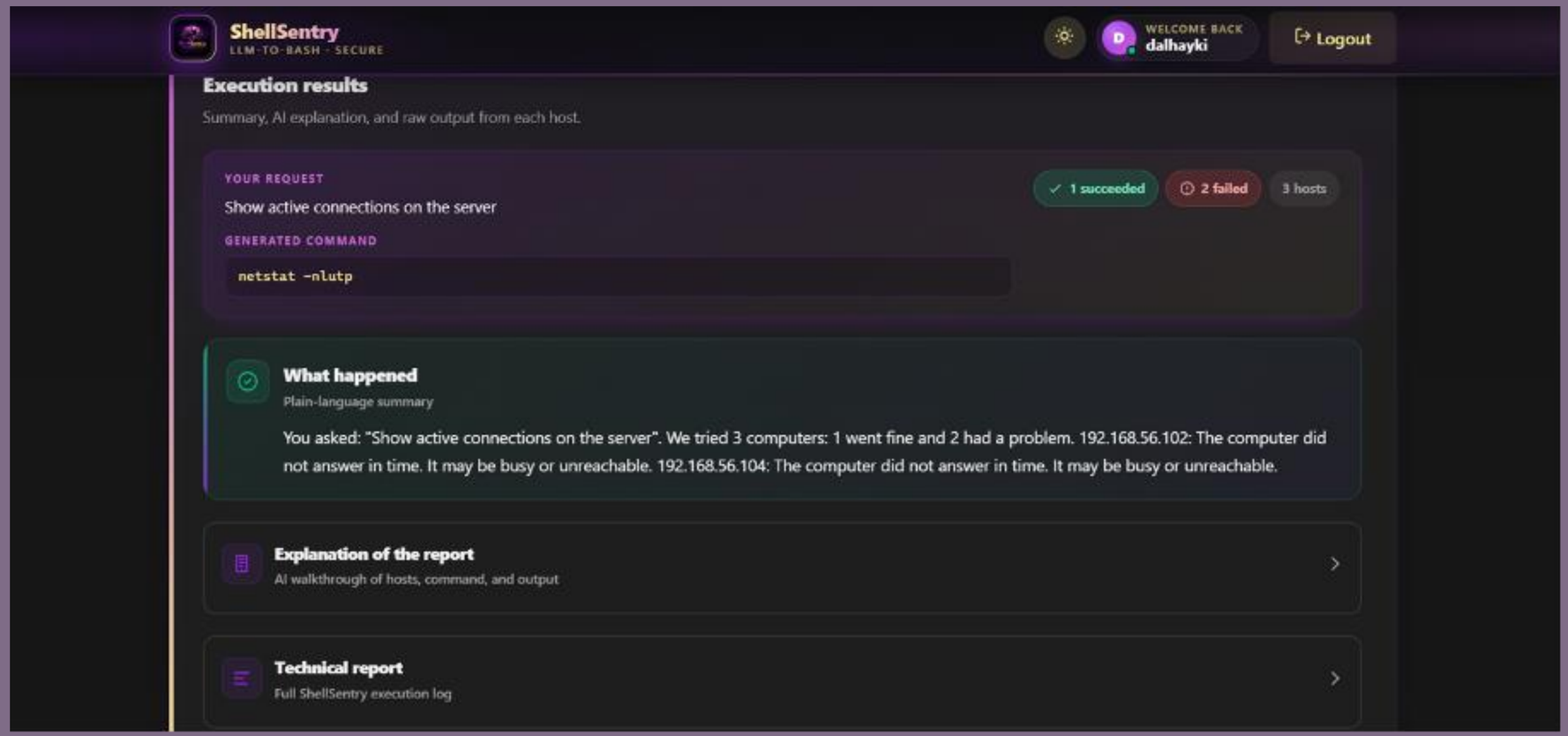
SYSTEM INTERFACE



Dashboard

Extras

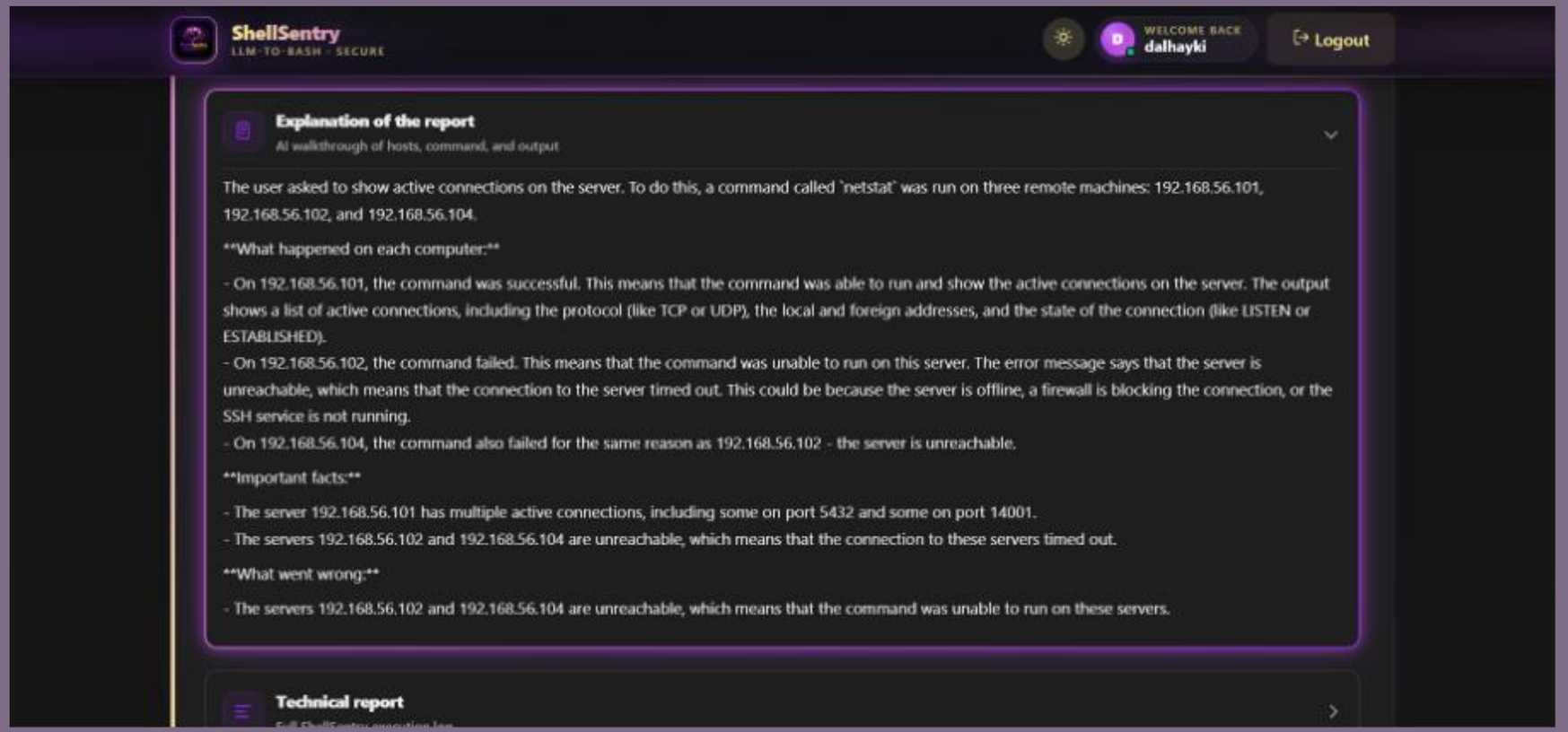
SYSTEM INTERFACE



Result page

Extras

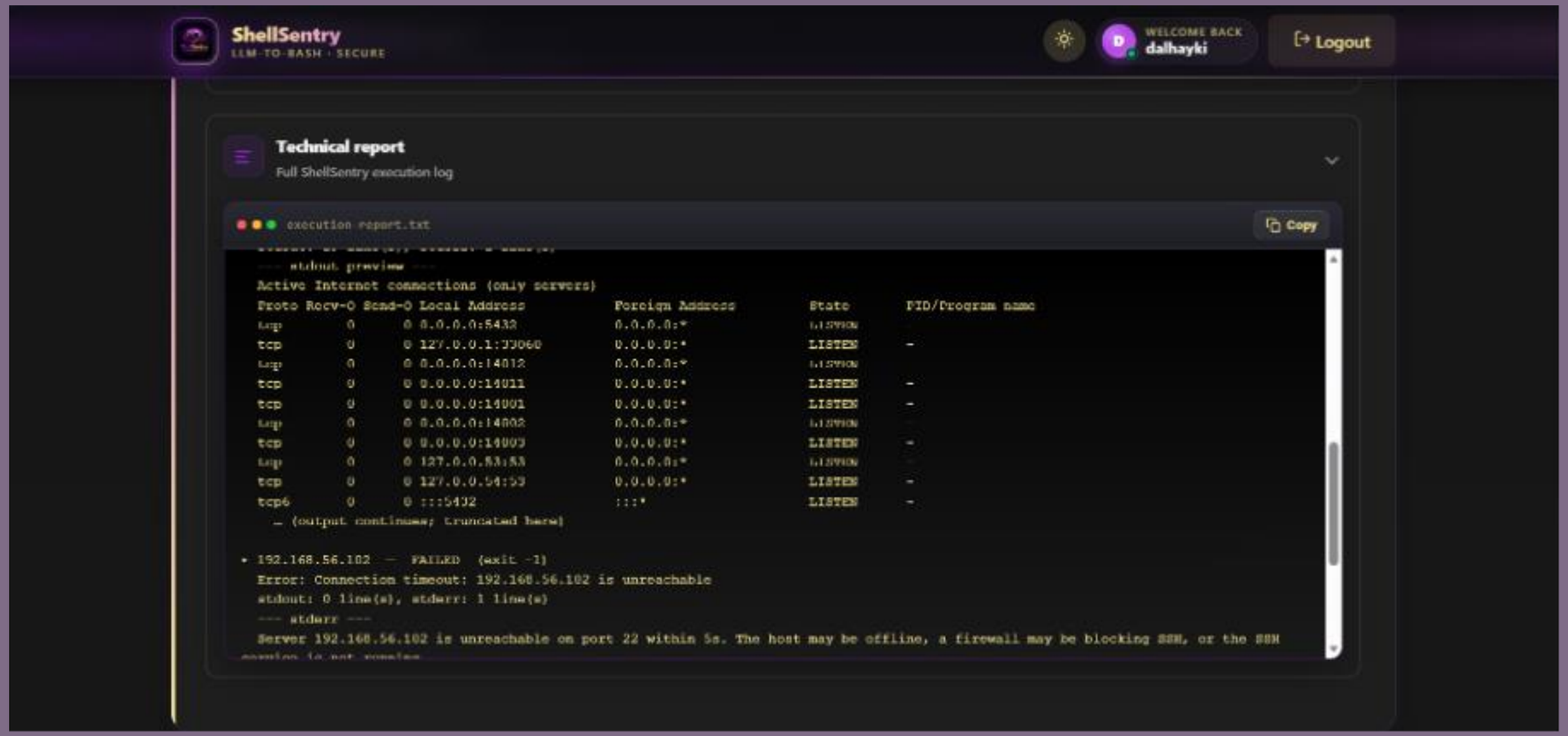
SYSTEM INTERFACE



Result page

Extras

SYSTEM INTERFACE



Result page

Extras