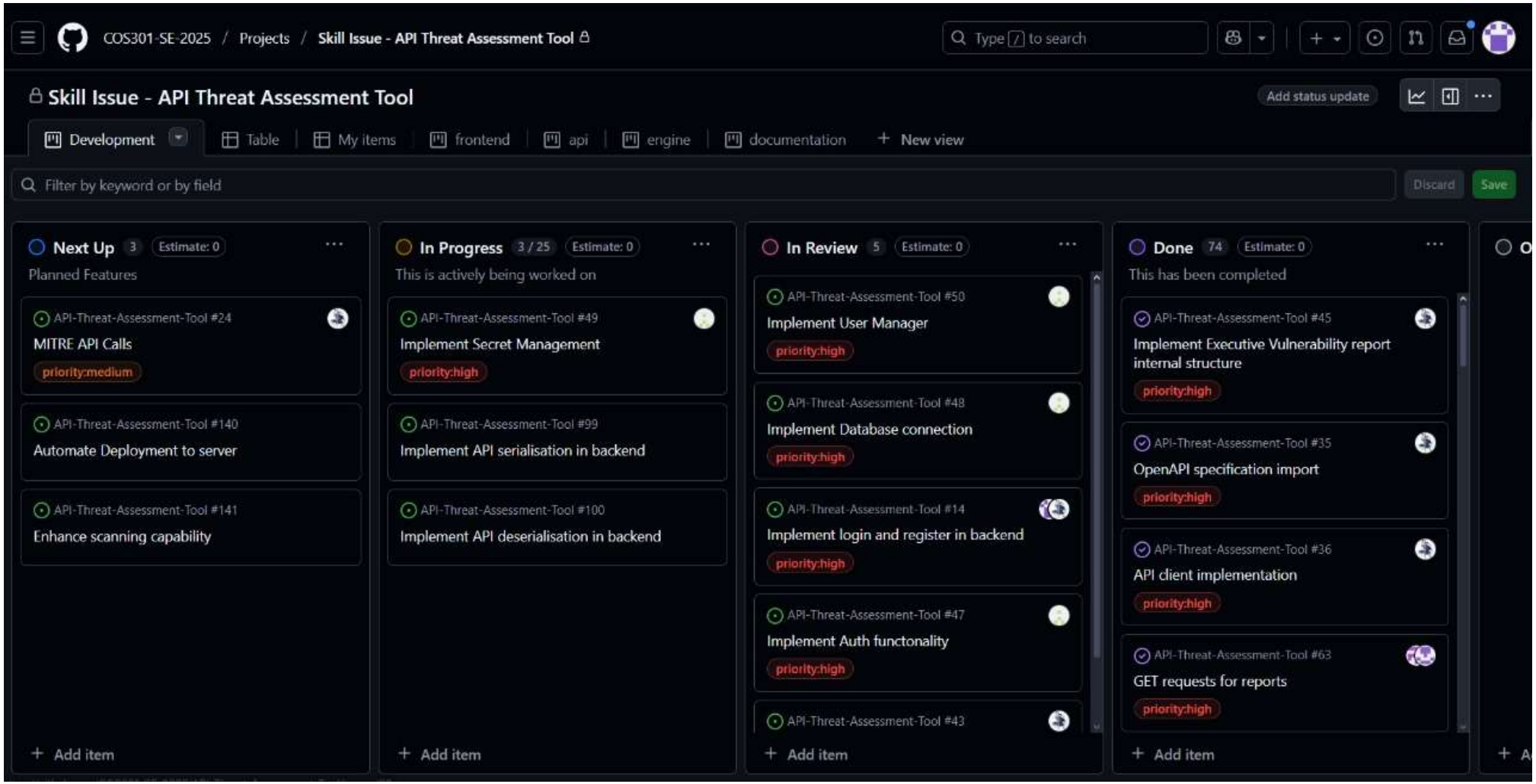
AT-AT

API THREAT ASSESSMENT TOOL

PROJECT DESCRIPTION

- API Threat Assessment Tool (AT-AT) is a **cybersecurity** application designed to automate the security testing of APIs. It enables organizations to identify vulnerabilities early, improve API resilience, and comply with industry security standards such as the OWASP API Security Top 10. The system will support specification-based scanning (e.g., OpenAPI, Postman Collections) to cover a wide range of API environments. Our team aims to build a scalable, extensible, and professional-grade API security assessment platform suitable for real-world deployment, allowing both automated and manual security evaluations.

PROJECT PROGRESS



Software Architecture

System Architecture Overview

The AT-AT system adopts a **microservices-inspired modular architecture** designed to support independent development, scalability, and maintainability. The system consists of clearly separated logical components, each responsible for a specific function.

Layers

1. Presentation Layer (Frontend)

- Built with React.js.
- Responsible for rendering the user interface.
- Sends requests to the API for scans, report generation, and user authentication.

2. API Layer (JS Express API)

- Acts as a middleware between frontend and backend processing.
- Performs user session validation, RESTful request parsing, and dispatches tasks to Python backend.

3. Processing Layer (Python Backend)

- Executes scanning logic, report generation, and vulnerability detection.
- Communicates with the API layer through internal HTTP calls.
- Reads environment variables for secure credentials.

Architectural Justification

The separation into distinct layers follows microservices principles to support:

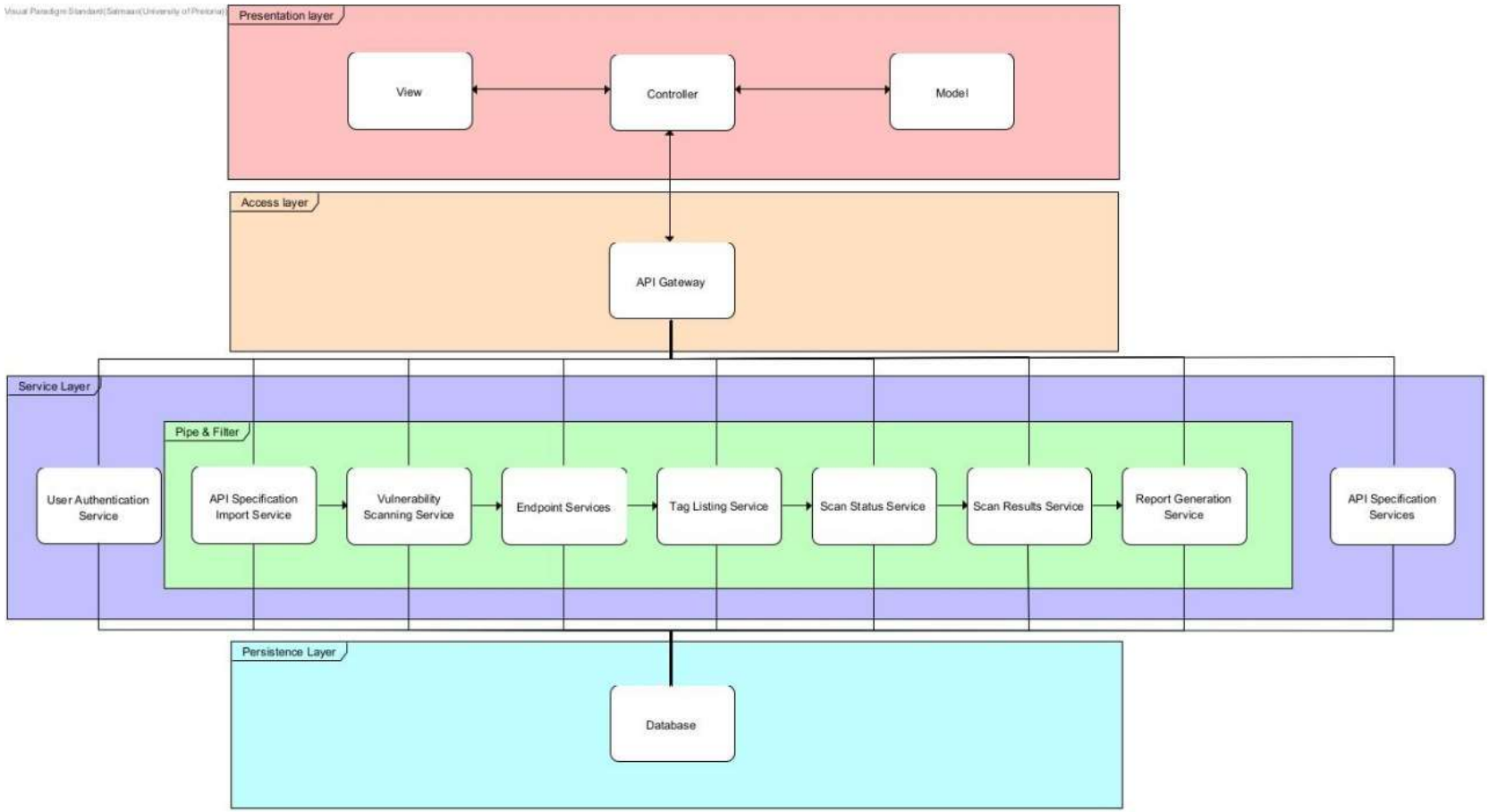
- **Scalability**: Each component (Frontend, API, Backend) can be deployed or scaled separately.
- **Maintainability**: Isolated codebases reduce complexity when updating or testing.
- **Fault Isolation**: Issues in one layer (e.g., backend processing) do not bring down the UI or authentication.

> Note: No direct communication occurs between the frontend and the backend processor — all traffic is routed through the API layer.

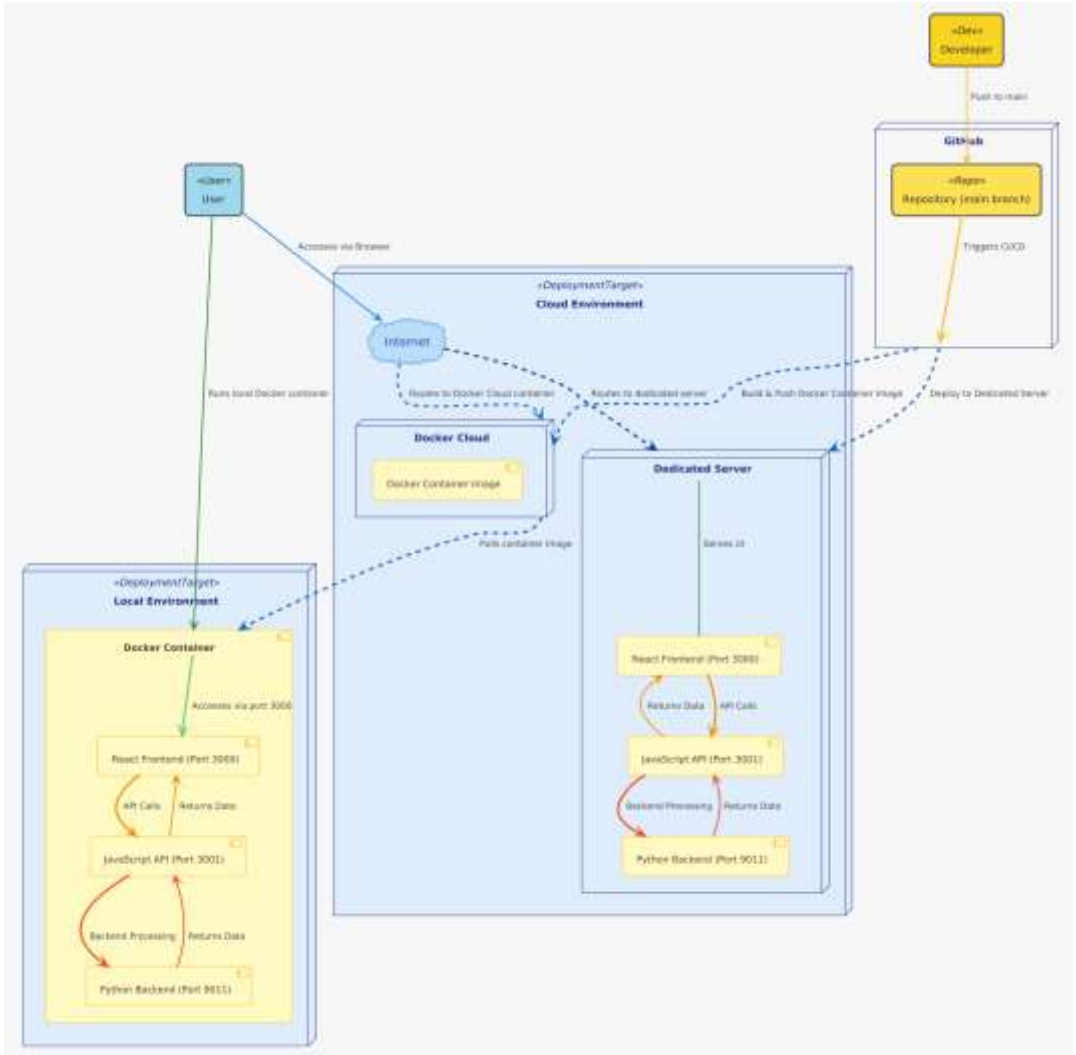
Non-Technical Notes

- This architecture is **technology-independent** in concept.
- It may be deployed to Docker, virtual machines, or dedicated servers, without altering the structural design.
- Quality Attributes (QAs) such as **performance**, **modularity**, and **reliability** are enhanced by this separation of concerns.

ARCHITECTURAL DIAGRAM



DEPLOYMENT MODEL



SERVICE CONTRACTS

AT-AT Service Contracts

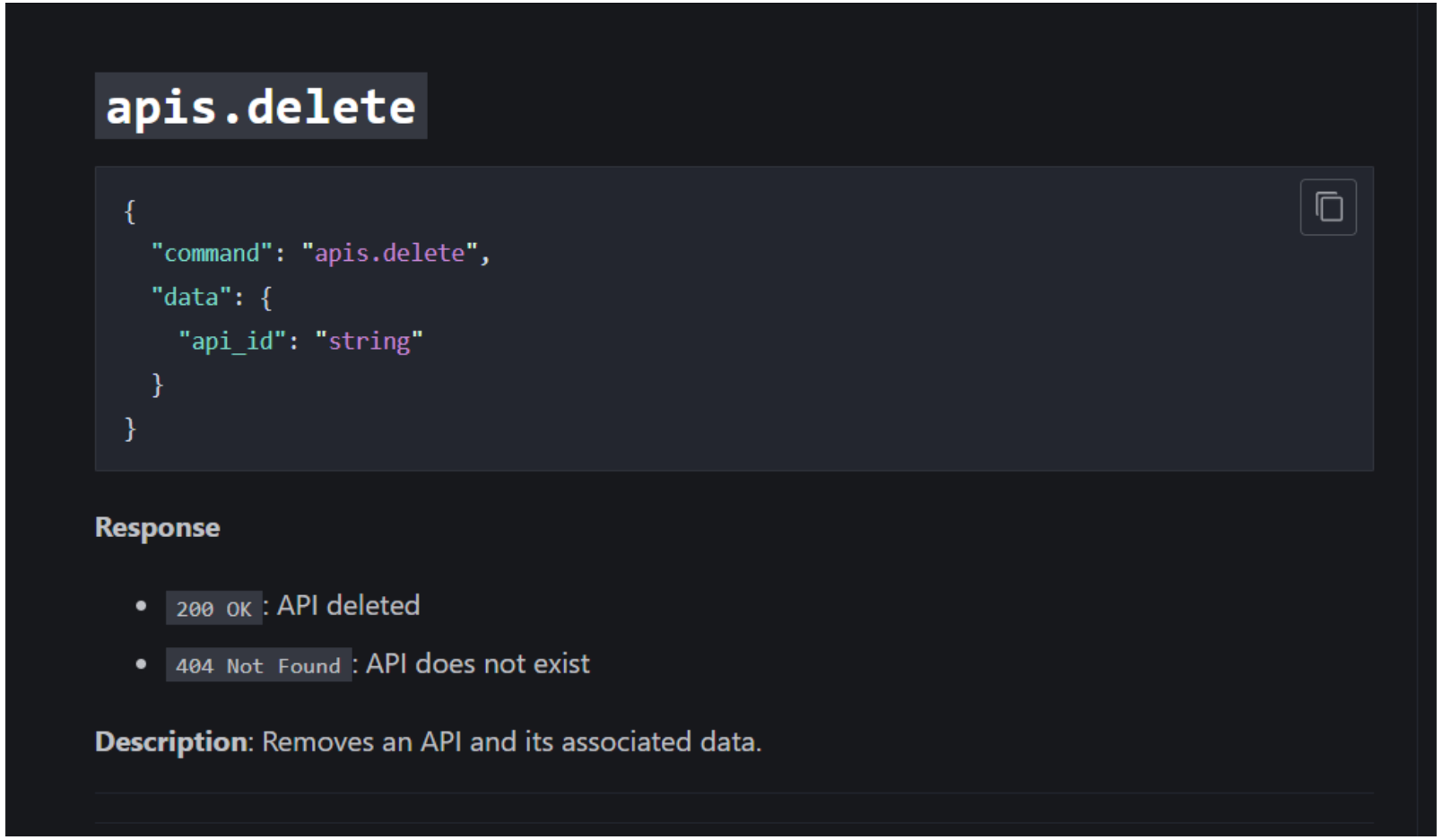
The API Threat Assessment Tool (AT-AT) uses a RESTful architecture and communicates using JSON over HTTP(S). The API is authenticated via API keys or Bearer tokens (for RBAC).

`apis.create`

```
{
  "command": "apis.create",
  "data": {
    "name": "string",
    "description": "string",
    "file": "OpenAPI spec file (encoded)"
  }
}
```

Response

SERVICE CONTRACTS



apis.delete

```
{
  "command": "apis.delete",
  "data": {
    "api_id": "string"
  }
}
```

Response

- `200 OK` : API deleted
- `404 Not Found` : API does not exist

Description: Removes an API and its associated data.

SERVICE CONTRACTS

Error Schema

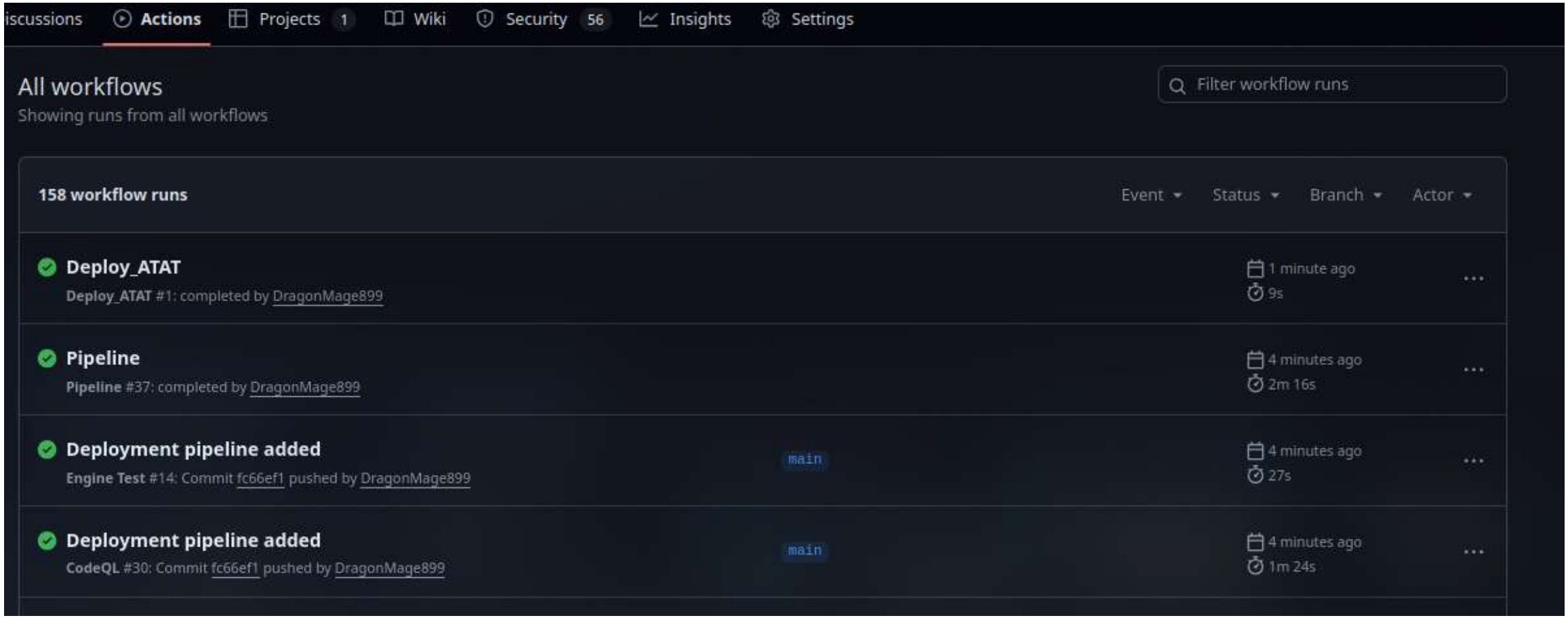
All endpoints return standardized error responses:

```
{
  "status": "error",
  "message": "Detailed error message"
}
```

Timeout & Limits

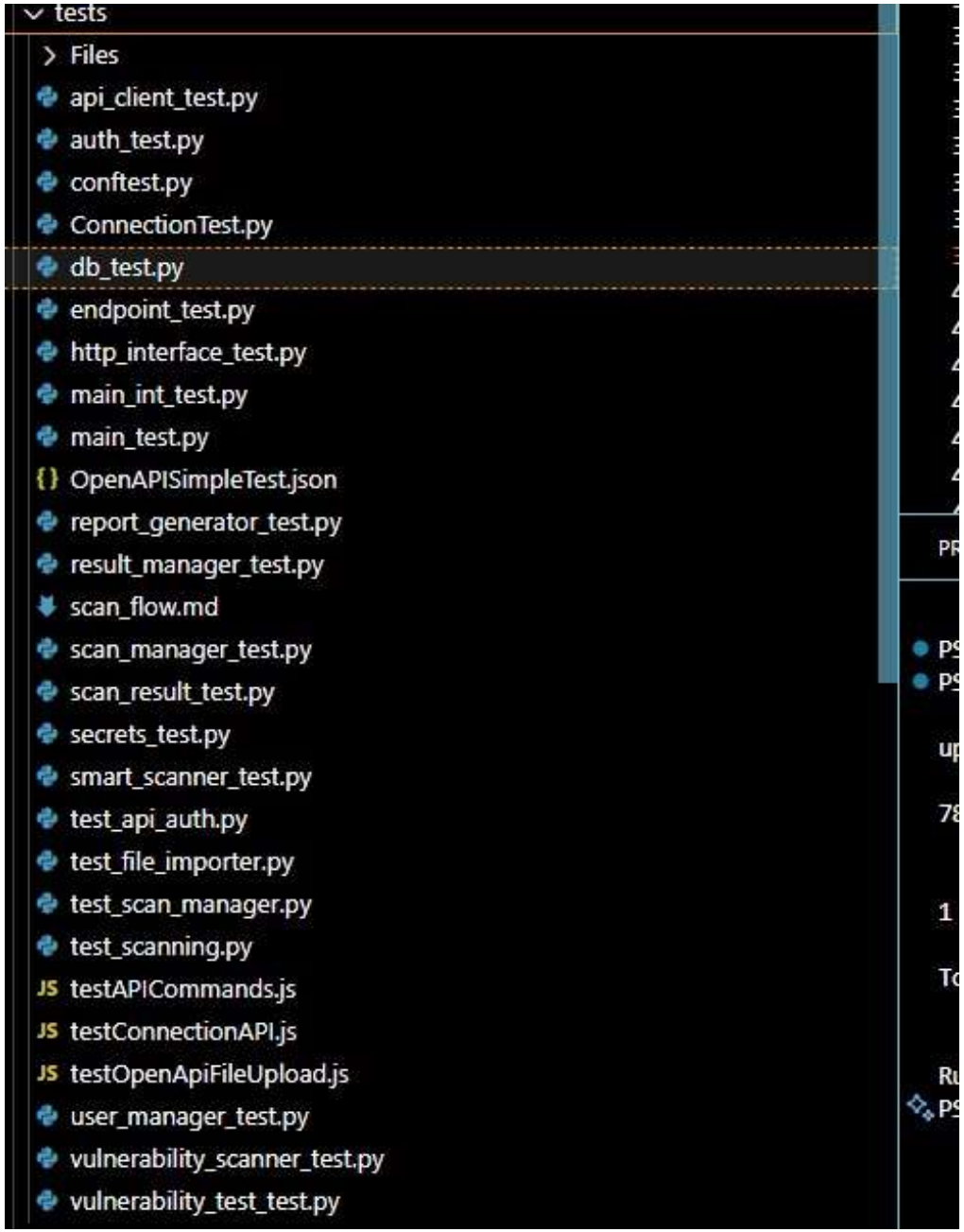
- All scan jobs are capped at 60 seconds.
- Endpoints are rate-limited to prevent abuse.

CI/CD

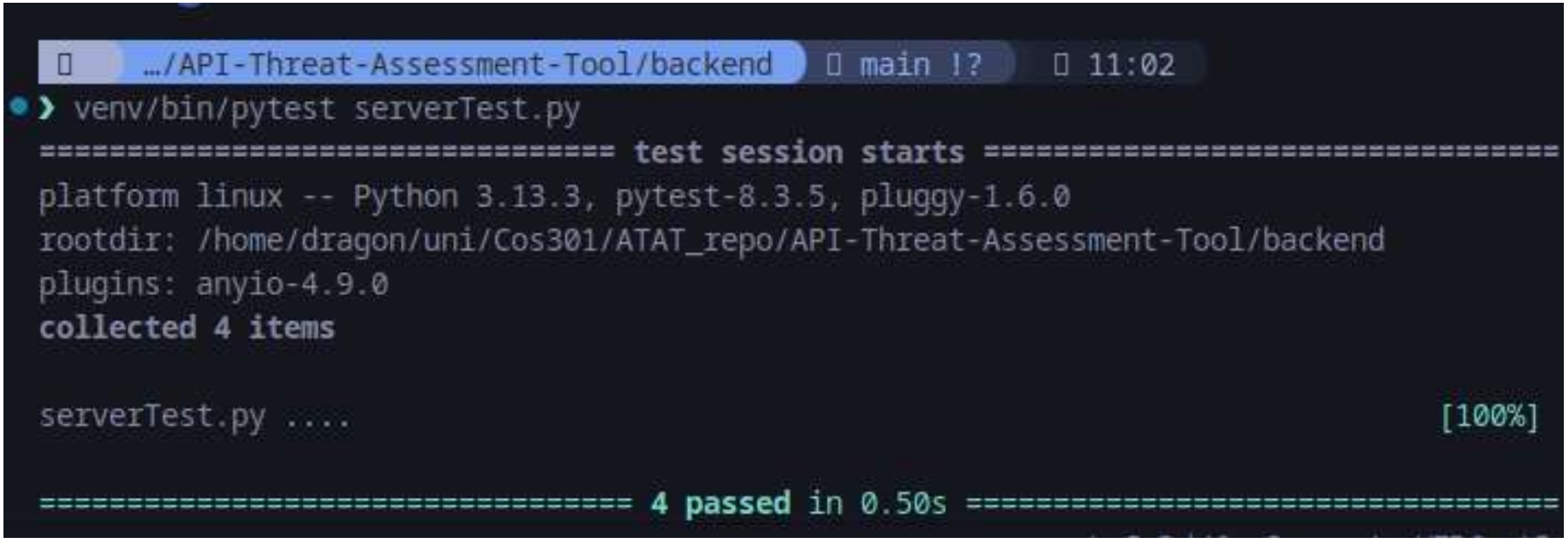

iscussions
Actions
Projects 1
Wiki
Security 56
Insights
Settings
All workflows
Showing runs from all workflows
Filter workflow runs
158 workflow runs
Event
Status
Branch
Actor
Deploy_ATAT
Deploy_ATAT #1: completed by DragonMage899
1 minute ago
9s
Pipeline
Pipeline #37: completed by DragonMage899
4 minutes ago
2m 16s
Deployment pipeline added
Engine Test #14: Commit fc66ef1 pushed by DragonMage899
main
4 minutes ago
27s
Deployment pipeline added
CodeQL #30: Commit fc66ef1 pushed by DragonMage899
main
4 minutes ago
1m 24s

TESTING…

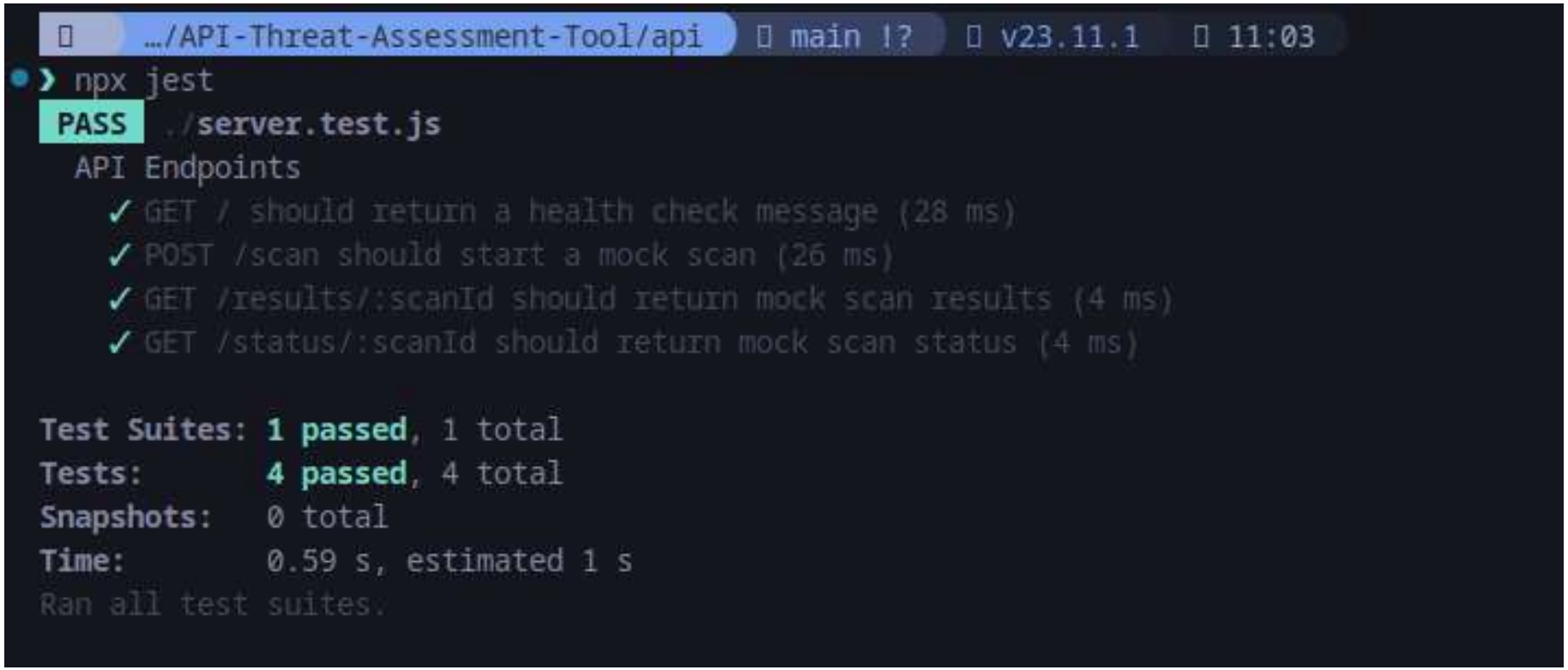

QA watching devs work on weekends after assigning critical bugs on friday evening.


tests
Files
api_client_test.py
auth_test.py
conftest.py
ConnectionTest.py
db_test.py
endpoint_test.py
http_interface_test.py
main_int_test.py
main_test.py
OpenAPISimpleTest.json
report_generator_test.py
result_manager_test.py
scan_flow.md
scan_manager_test.py
scan_result_test.py
secrets_test.py
smart_scanner_test.py
test_api_auth.py
test_file_importer.py
test_scan_manager.py
test_scanning.py
testAPICommands.js
testConnectionAPI.js
testOpenApiFileUpload.js
user_manager_test.py
vulnerability_scanner_test.py
vulnerability_test_test.py

TESTING BACKEND



TESTING API


…/API-Threat-Assessment-Tool/api
main !?
v23.11.1
11:03
❯ npx jest
PASS ./server.test.js
API Endpoints
✓ GET / should return a health check message (28 ms)
✓ POST /scan should start a mock scan (26 ms)
✓ GET /results/:scanId should return mock scan results (4 ms)
✓ GET /status/:scanId should return mock scan status (4 ms)
Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 0.59 s, estimated 1 s
Ran all test suites.

TESTING FRONTEND

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	63.95	46.77	68.57	64.58	
AuthContext.js	70.1	56.25	75	72.52	9,51,125-129,159-187,197-198,203-204
Construction.js	7.14	0	0	7.14	8-57
Dashboard.js	4.34	0	0	4.34	8-137
Home.js	100	50	100	100	29-90
Login.js	76.47	85.29	83.33	78.12	54,61-89,110
ManageAPIs.js	56.17	45.31	63.63	55.81	48-51,57,146-194,284,363-367
Signup.js	78.12	76.47	100	78.68	39-40,43-44,51-52,55-56,59-60,63-64,101

```
Test Suites: 8 passed, 8 total
Tests:       44 passed, 44 total
Snapshots:   0 total
Time:        12.388 s
Ran all test suites.
```

QUESTIONS

THANK YOU

SKILL ISSUE