|  |  |  |  |
| --- | --- | --- | --- |
| User Story | Acceptance Criteria | Developer Tasks | Time Estimate |
| **As a DevOps Engineer**, I want to send system logs to SentinelAI via an API, so that I can track application activity in a centralized and secure system. | . A POST /logs API endpoint exists  . It accepts fields like: username, ip\_address, action, result, endpoint  .Valid logs are stored in PostgreSQL  . Invalid data is rejected with a meaningful error |  Design LogCreate schema using Pydantic   Create LogModel using SQLAlchemy   Implement POST /logs endpoint in FastAPI   Add database logic to save logs   Test log insertion with valid and invalid inputs | 2 Days |
| **As a Security Analyst**, I want to retrieve all submitted logs from SentinelAI via an API, so that I can review system activity and detect anything unusual. |  A GET /logs endpoint is available   The endpoint returns a list of stored logs   Each log includes readable: username, ip\_address, action, result, endpoint, and timestamp   Response is structured and clean (JSON) |  Create LogOut schema using Pydantic for response formatting   Implement GET /logs endpoint in FastAPI   Fetch logs from the PostgreSQL database   Return logs in proper JSON format | 1 Day |
| **As a Security Analyst**, I want to verify that only valid logs are accepted by SentinelAI, so I can be confident that the stored data is clean and reliable. |  The API rejects logs with missing or invalid fields   The system gives meaningful error responses (e.g. 400/422 errors)   Only valid logs are saved to the database |  Validate incoming data with Pydantic   Add error messages for missing/invalid fields   Test behavior with bad and good inputs   Confirm only valid logs are stored | 1 Day |
| **As a System Admin**, I want SentinelAI to return clear and meaningful error messages when something goes wrong, so that I can quickly identify and fix issues in production. |  If a log submission fails (e.g., missing field or DB issue), the API responds with a clear error message   The error includes status code (400, 422, 500) and explanation   Errors are logged for later review |  Implement proper error handling using FastAPI’s HTTPException   Catch database or validation errors with try/except   Log the error details (optional: use logging module)   Return appropriate error codes and messages | 1 Day |
| **As a Security Analyst**, I want the log data returned by SentinelAI to include timestamps and user actions in a clear format, so I can trace when and how suspicious activities occurred. |  Each log entry includes: username, ip\_address, action, result, endpoint, timestamp   Timestamps are formatted in UTC or readable ISO format   The log list returned from /logs is clean and consistent |  Ensure the timestamp field is stored and returned properly   Format the timestamp using datetime.utcnow() and Pydantic   Clean up JSON response structure in the GET route   Confirm that every log entry includes all key fields | 1 Day |
| **As a DevOps Engineer**, I want clear setup instructions and an organized folder structure for SentinelAI, so I can deploy or extend the system without confusion. |  A README.md exists with project setup steps, description, and usage instructions   Folder structure is documented in a PROJECT\_STRUCTURE.md file   Environment setup (e.g., .env, database) is clearly explained |  Write a clear and professional README.md including:   * Project description * Installation steps * Running instructions * API usage summary    Create PROJECT\_STRUCTURE.md showing backend folder layout   Document .env usage and config in the README | 1 Day |
| **As a System Admin**, I want to ensure that only clean environment variables are used in SentinelAI, so I can manage credentials and deployment settings securely. |  The database URL and other sensitive config values are read from .env   No secrets or connection strings are hardcoded anywhere   The backend runs using environment-based configuration only |  Create .env file and define DATABASE\_URL   Load environment variables using python-dotenv   Update config.py to use .env values   Remove any hardcoded credentials from the project | 0.5 Day |
| **As a QA Analyst, I want SentinelAI’s log API to be tested automatically, so I can be confident that core features work reliably even after future changes.** |  Automated tests exist for:   * Submitting valid logs (✅ should store) * Submitting invalid logs (❌ should reject) * Retrieving logs (✅ should return stored entries)    Tests can be run using pytest or FastAPI TestClient   Test results are clearly shown in the terminal |  Create a tests/ folder with clean structure   Write test cases for:   * Valid POST /logs * Invalid POST /logs (e.g., missing field) * GET /logs returns correct structure    Use TestClient from FastAPI or pytest to run tests   Document how to run tests in the README.md | 1 Day |