Faculty of Engineering, Science & Technology

IQRA UNIVERSITY



FYP – 1: Architecture Report

[Spring-2025]

**Course Instructor:**

Dr Aarij Hussain

**Supervisor:**

Dr Muhammad Naveed

**Group Members:**

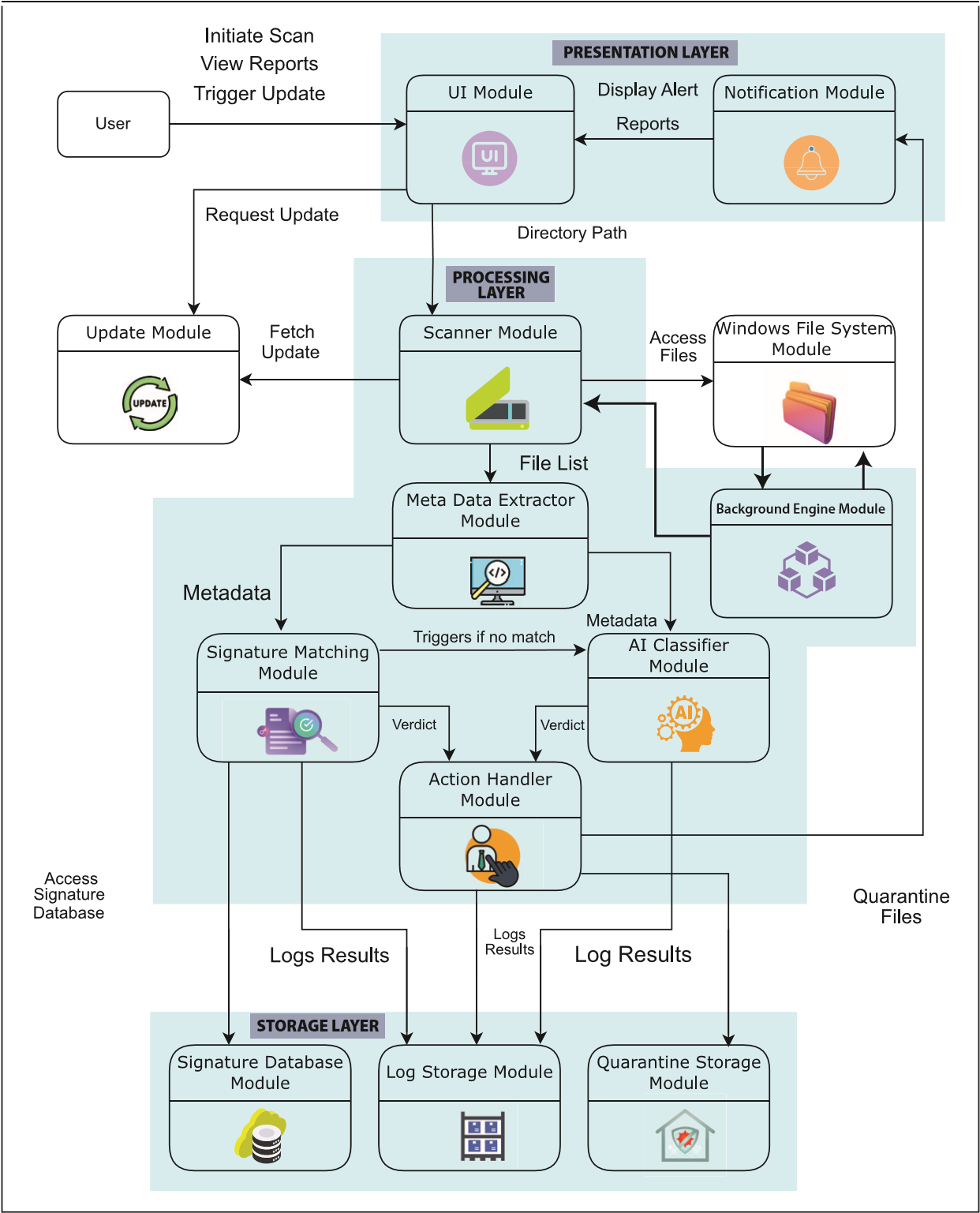
Muhammad Usman (IU05-0122-0303)

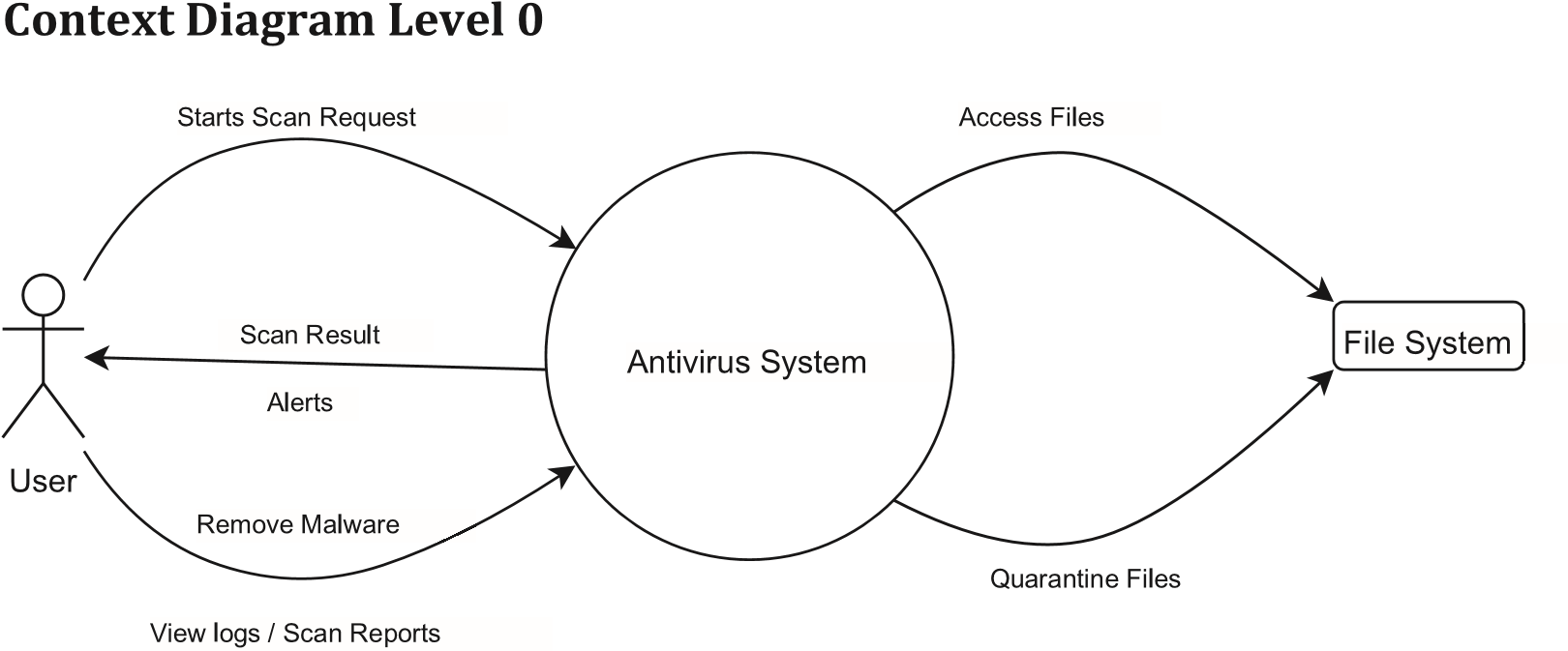
Bin Yameen (IU05-0122-0303)

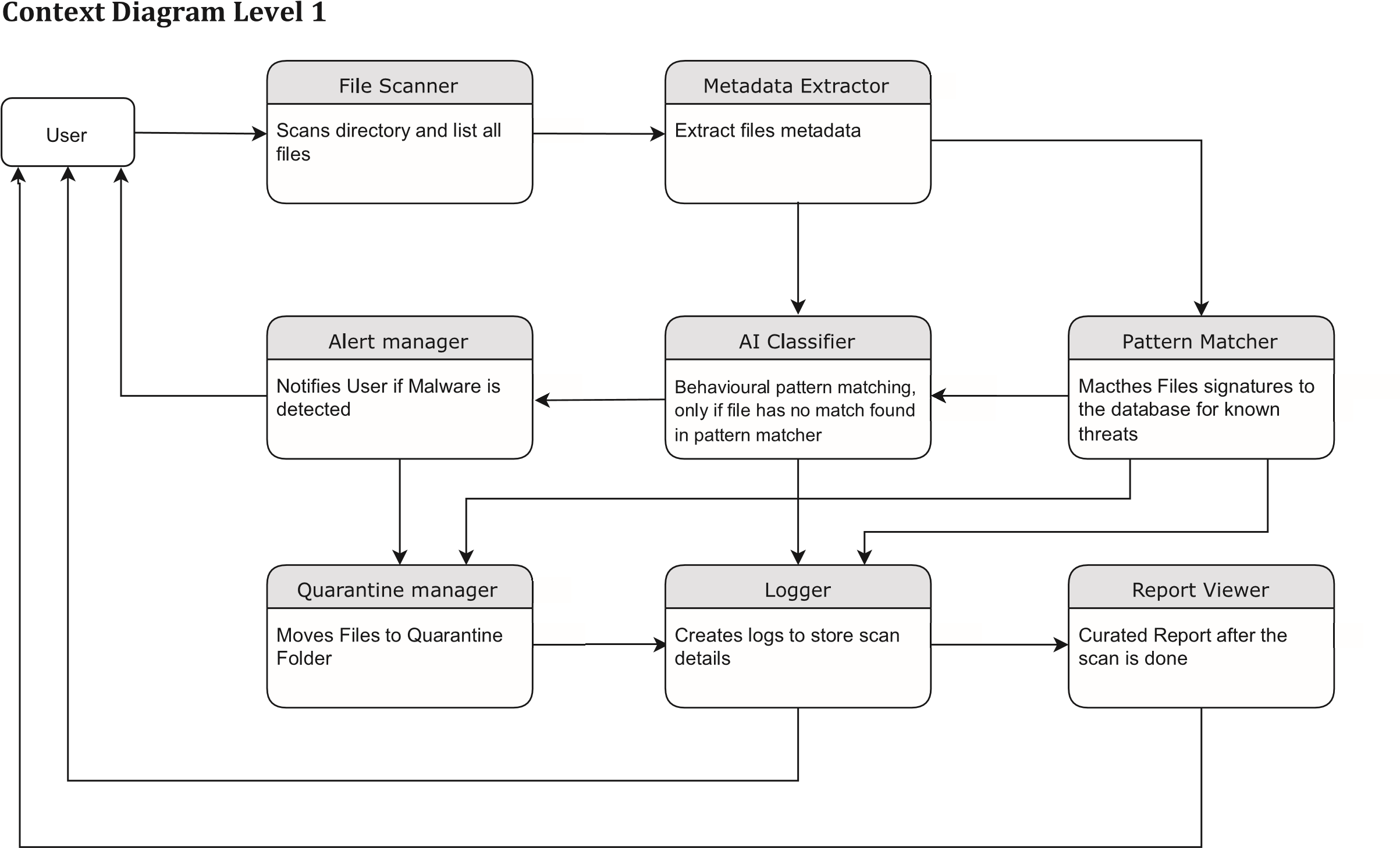
Syed Muhammad Daniyal (IU05-0122-0497)

**Submission Date:**

19th - June - 2025







**Technology Stack**

* **Programming and Scripting Languages**

|  |  |
| --- | --- |
| **Language** | **Purpose** |
| **C** | Core antivirus engine, scanner, detection logic |
| **Python** | AI model development and classification |
| **Batch / Shell Scripting** | Automation (launch, logging, updates) |

* **Libraries**

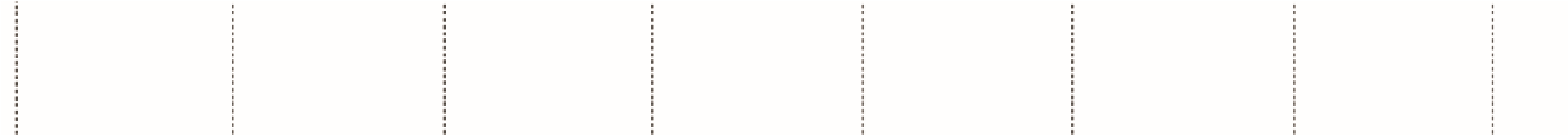
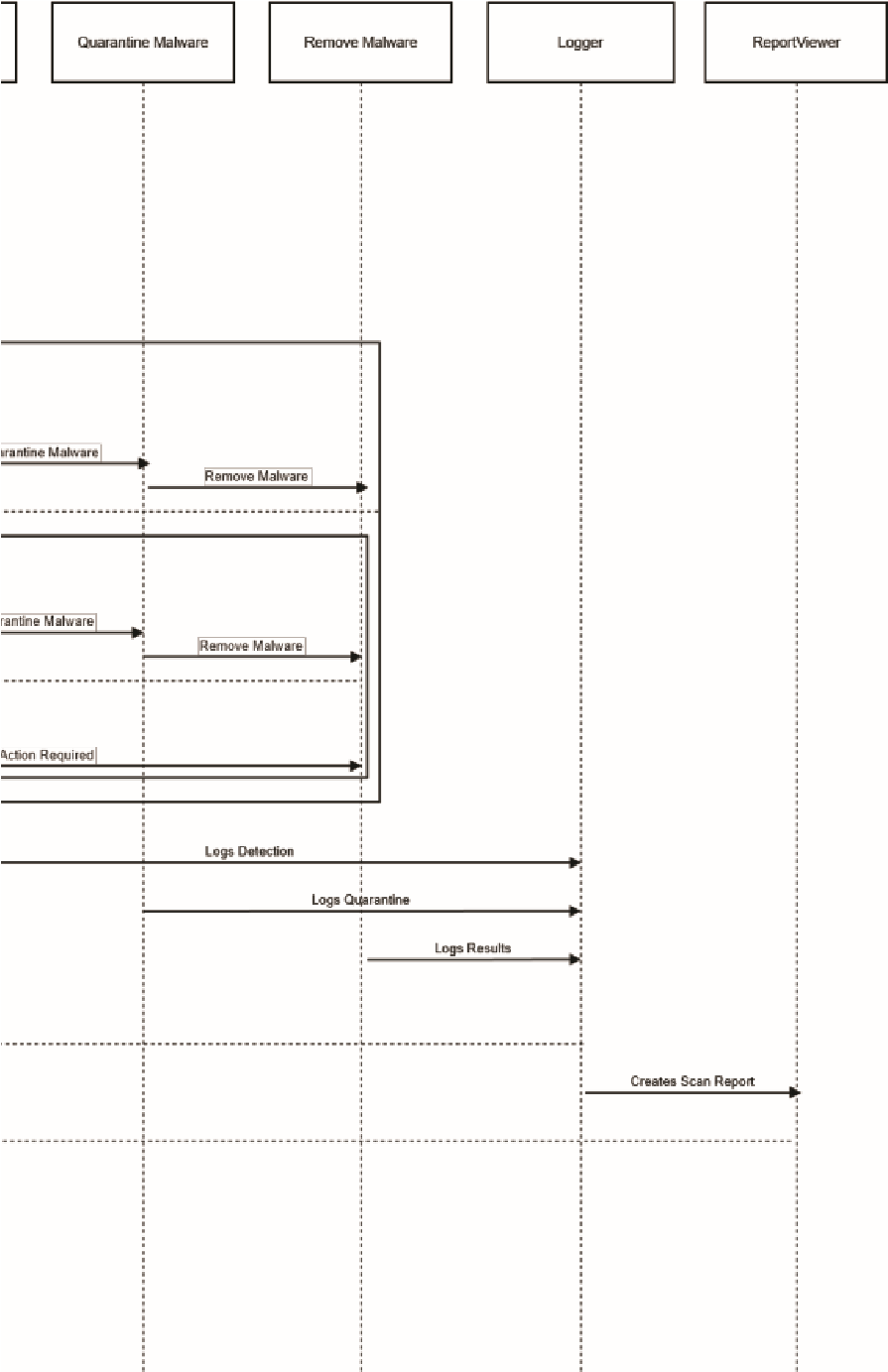
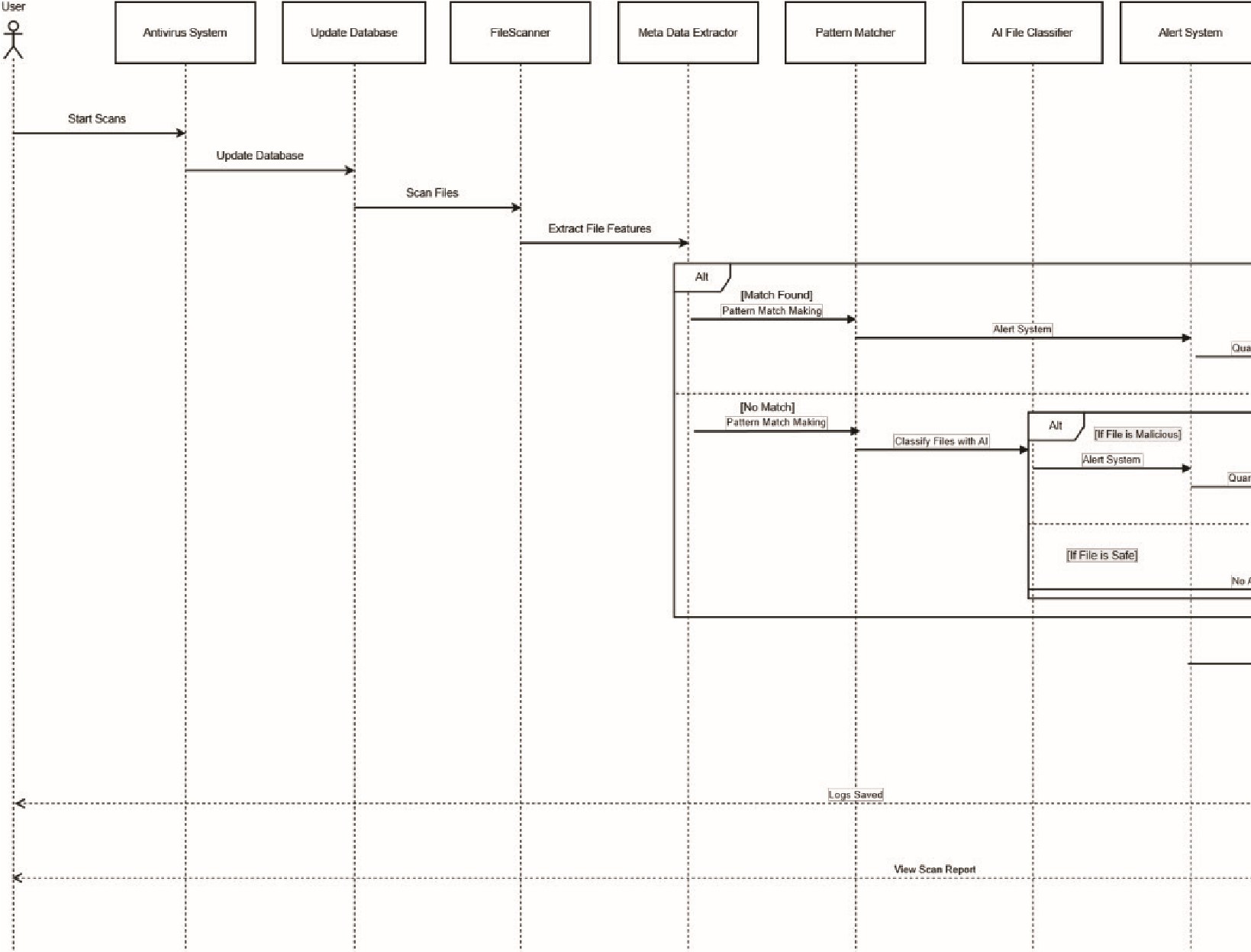
|  |  |
| --- | --- |
| **Library** | **Purpose** |
| **Standard C Library** | stdio.h, stdlib.h etc. for core logic |
| **Windows API (via headers)** | File operations, permissions, directory watching for real-time scanning |
| **OpenSSL** (libssl, libcrypto) | hash computation |

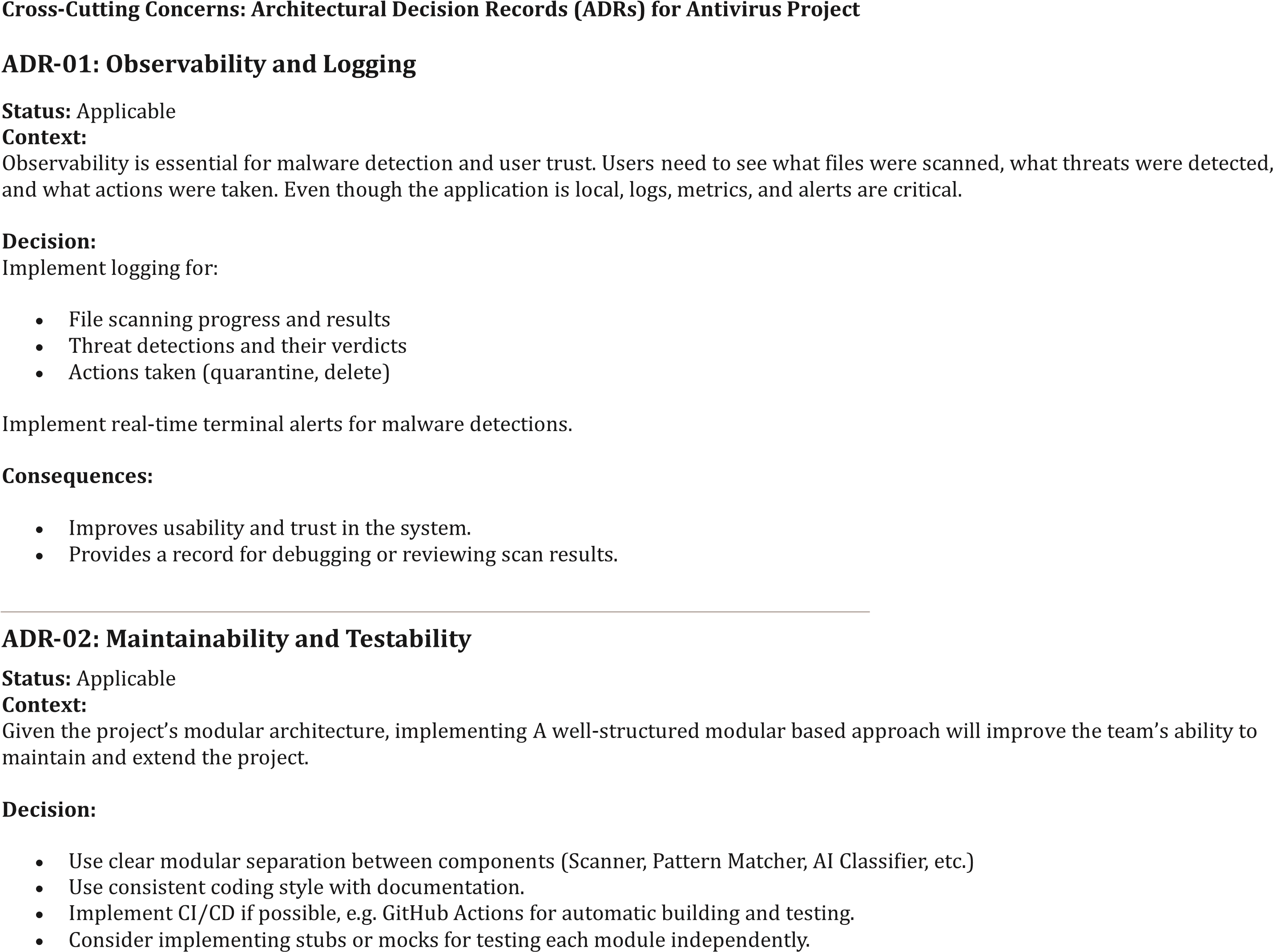
* **Tools**

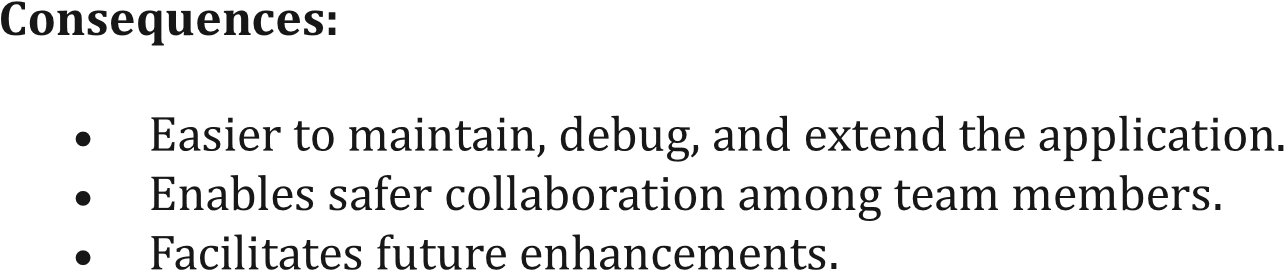
|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| **VS Code** | Main development IDE |
| **MSYS2 + MINGW64** | GCC compiler |
| **GCC (via MINGW64)** | Compile C code |
| **Git** | Version control (local repo) |
| **GitHub** | Code backup, collaboration, issue tracking |
| **PowerShell / MINGW64 Terminal** | testing CLI output |
| **Python 3.x** | Running AI models and preprocessing |

* **Database**

|  |  |
| --- | --- |
| **Database** | **Purpose** |
| **CSV or .JSON** | For Storing Crypographic Hashes, Reggex Pattern and logs |







**Compliance**

1. **Follows Standard Antivirus Design Pattern**  
   The system adheres to conventional antivirus workflows — including scanning, detection (signature-based and behavioral), alerting, logging, and optional quarantine.
2. **Supports Offline Detection Mechanisms**  
   The system functions fully offline using local signature and hash databases — enabling scans without cloud or network dependencies.
3. **Layered Architecture & Modular Implementation**  
   Uses a modular structure (scanner, detector, logger, etc.) for maintainability and testability. Logic is separated from UI and storage.
4. **Safe Handling of Malicious Files**  
   The antivirus does not auto-delete files without logging or user consent for quarantine or removal. User control is preserved.
5. **Structured Logging and Transparency**  
   All scan results, detections, and actions are logged in readable formats (.json, .csv) to support auditing and traceability.
6. **AI Classification Compliant with Model Transparency**  
   ML-based classification decisions are logged and auditable, with features stored for each scan decision.

**Constraints**

1. **Platform-Specific**  
   Developed and tested exclusively for Windows 10/11 environments using MINGW64.
2. **Resource Constraints for AI**   
   AI models are CPU-bound and lightweight. No GPU acceleration or large-scale datasets involved due to hardware limits.