National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Roshan Muttath Francis
Student ID: X23299401

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	ROSHAN MUTTATH FRANCIS
Student ID:	X23299401
Programme:	MSc Cybersecurity **Year:** 2024-2025
Module:	MSc Research Project
Supervisor:	Michael Prior
Submission Due Date:	12/08/2025
Project Title:	Hybrid Browser-Based Framework for Mobile Threat Detection and Forensic Analysis
Word Count:	1133 **Page Count:** 13

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ROSHAN MUTTATH FRANCIS

Date: 12/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Hybrid Browser-Based Framework for Mobile Threat Detection and Forensic Analysis

ROSHAN MUTTATH FRANCIS
Student ID: x23299401

1 APP Configuration

1.1 Environment Configuration

- Before running you should create a .env file on backend folder for database connection:

```
```env
DATABASE_URL="postgresql://username:password@localhost:5432/your_database_name?sche
ma=public"
"
```

1.2 Database Setup steps:

```
npx prisma migrate dev --name init
npx prisma generate
```

1.3 Backend Server Startup:

```
cd backend
npm start
```

1.4 To run FastAPI Server:

(Linux/MacOS):

```
cd backend/fastapi
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
uvicorn whatsapp_api:app --reload --host 0.0.0.0 --port 8000
```

(windows):

```
powershell
cd backend/fastapi
python -m venv venv
.\venv\Scripts\Activate.ps1
pip install -r requirements.txt
uvicorn whatsapp_api:app --reload --host 0.0.0.0 --port 8000
```
```

1.5 Frontend Server Startup:

```
cd frontend
npm run dev
```

- Backend running on:

```
http://localhost:4000
```

- frontend server running on:

```
http://localhost:3000
```

1.6 Environment variable Configuration and installation for Aleapp and:

```
cd backend/ALEAPP/
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

2 PWA Setup (Offline-First Forensic Platform)

The Progressive Web App (PWA) is configured for cross-platform compatibility, enabling local forensic analysis without server dependency (Mobile Forensic, 2023).

Steps:

- Host PWA files (HTML5, CSS3, JS) in a browser-compatible format.
- Enable Service Worker for offline-first operation.
- Test compatibility with Chrome, Firefox, and Edge.

Purpose: Ensures secure, portable forensic tool accessible in field conditions.

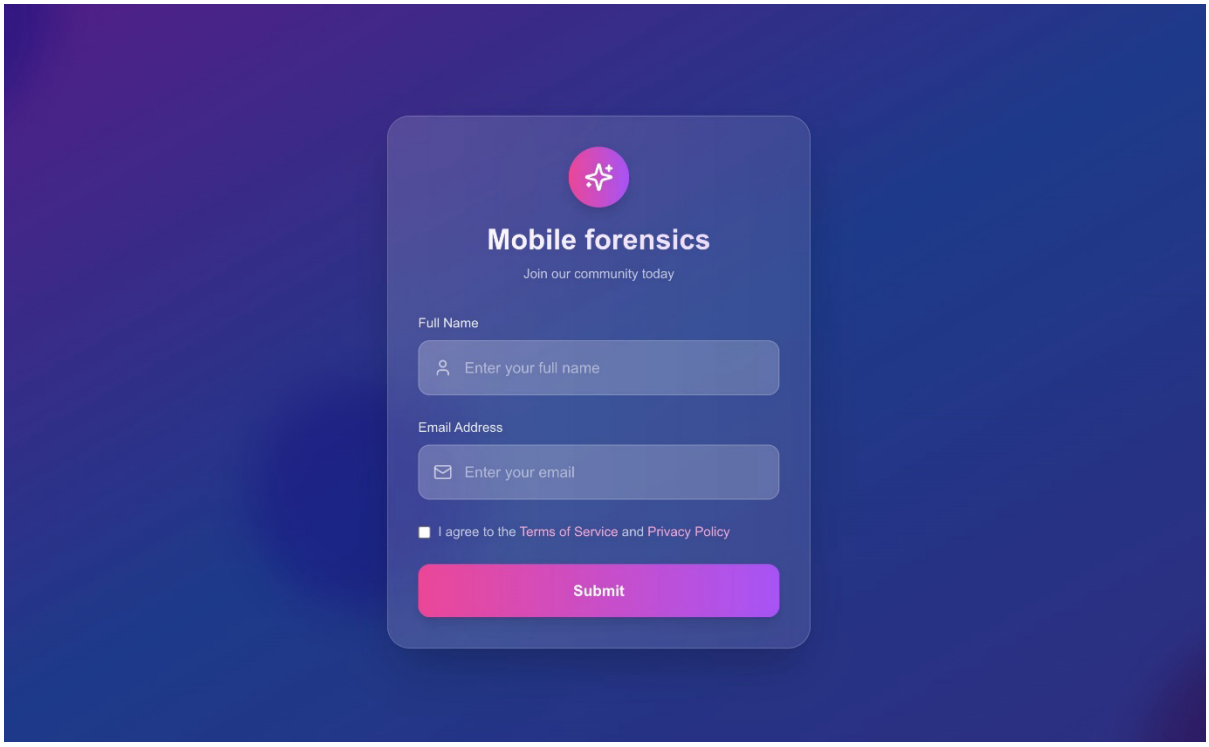

Figure 1: Login page

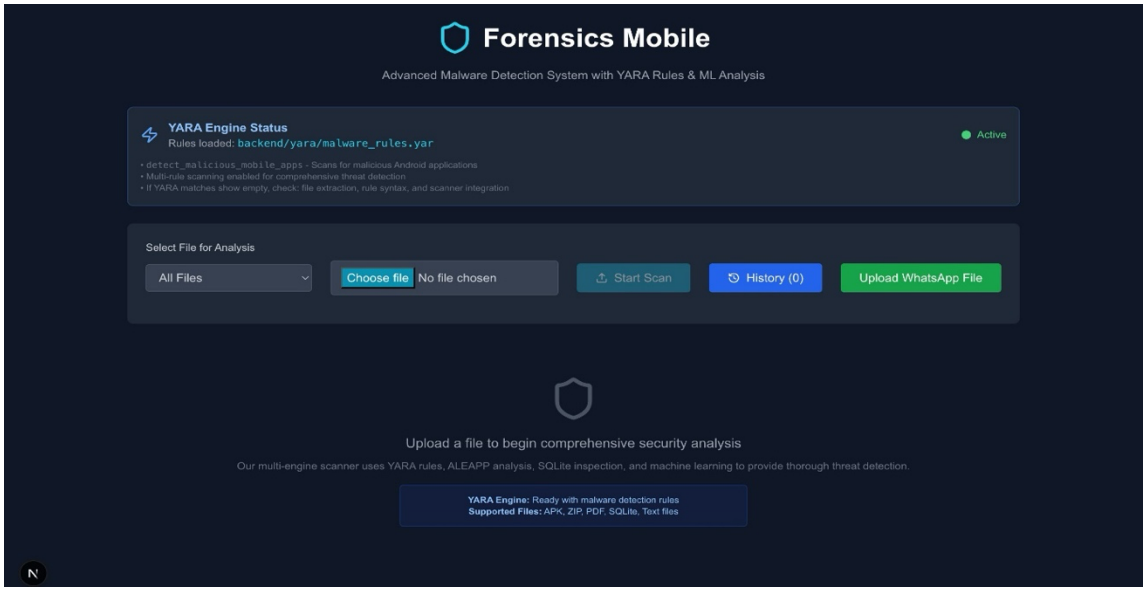

Figure 2: Uploading File Page

3 YARA Rules Configuration

YARA Loader Setup: (Brassard-Gourdeau and Khoury, 2020)

- Load Advanced_Malware_Generic, RAT_Remote_Access_Advanced, Ransomware_Locker_Advanced, and Downloader_Malware_Advanced.
- Allow custom rule uploads for specific investigations.
- Test with known malware samples for detection accuracy.

Result: Confirmed detection of multiple threats with correct rule matching.

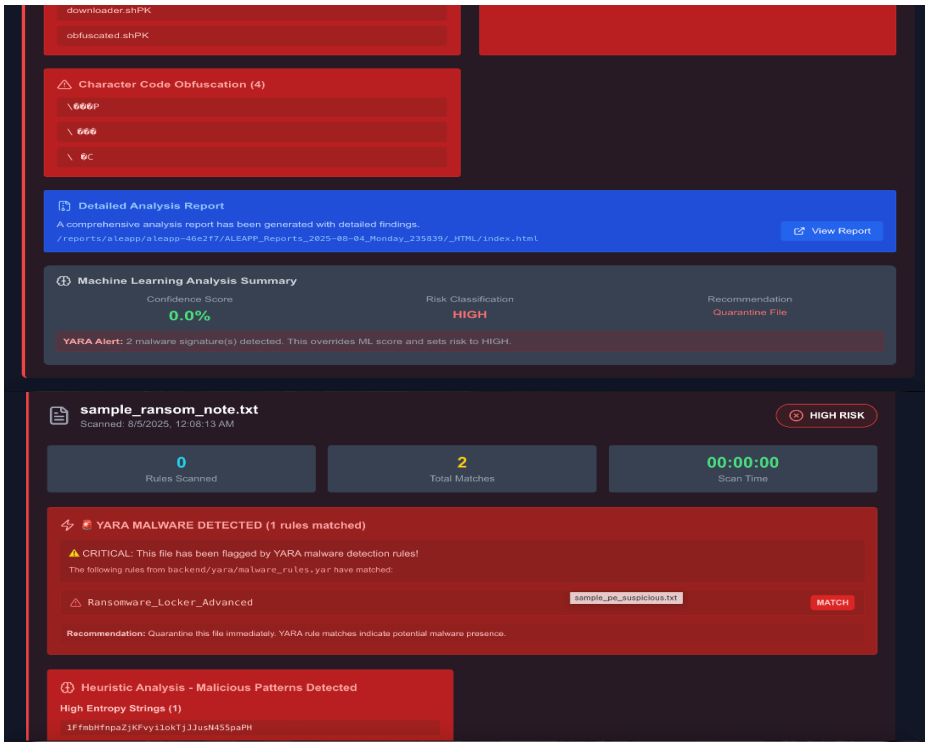

Figure 3: Malware detection

4 Regex Pattern Manager

Configuration:

- Load regex patterns for IoCs (URLs, IPs, emails, abusive terms).
- Allow investigators to add/edit patterns dynamically.
- Validate regex execution speed and pattern accuracy.

Result: Detected phishing URLs and obfuscated strings during malware scans (Gupta *et al.*, 2024).

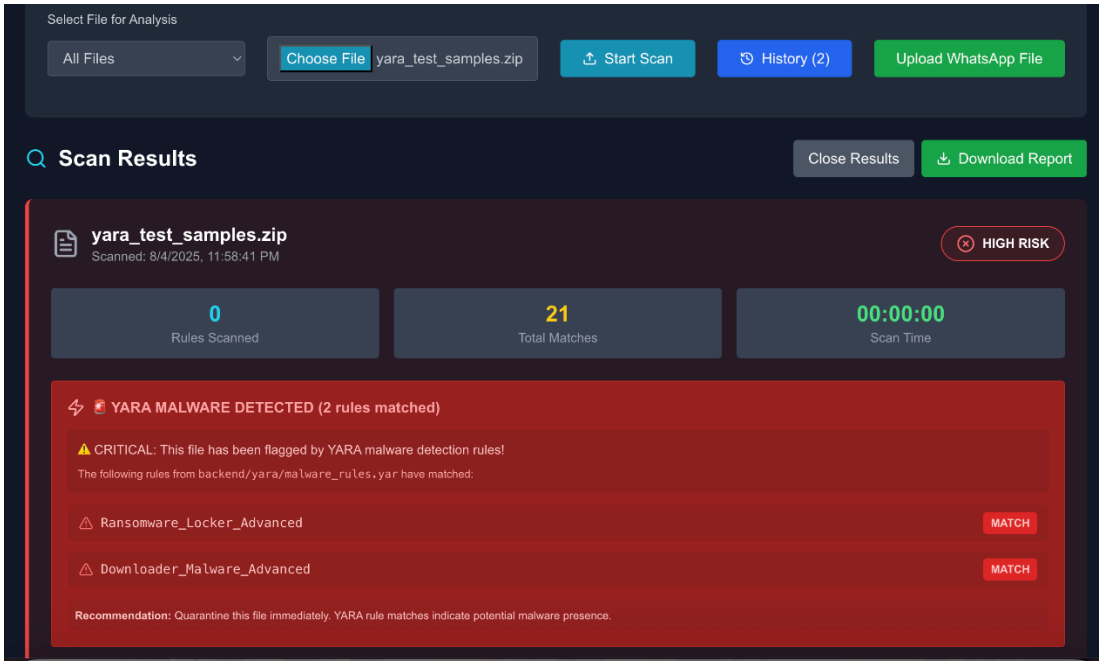

Figure 4: Scan Results

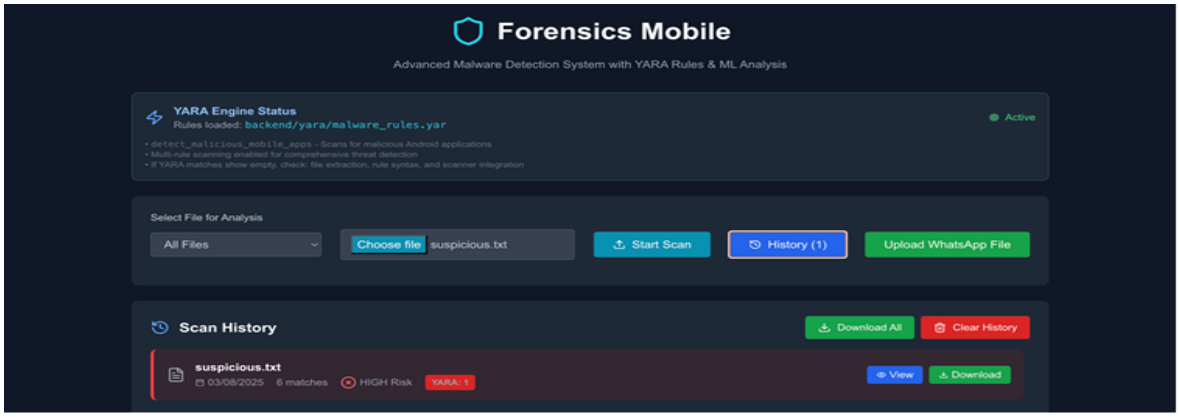

Figure 5: Scan History

The model is developed using Python and exported as a Keras model for deployment. It is integrated into a FastAPI backend, also built with Python, to handle inference requests. A Next.js frontend application communicates with the FastAPI server via API calls. This setup enables real-time interaction, where WhatsApp chat data is sent from the frontend to the backend, allowing the model to perform chat analysis and return the results seamlessly through the API.

5 WhatsApp RNN Threshold Settings

Configuration:

- Upload RNN model for toxicity detection.
- Set classification thresholds for Low, Medium, High severity.
- Test using .txt and msgstore.db datasets.

Result: Achieved **91% accuracy** on test dataset; toxic messages flagged with severity labels.

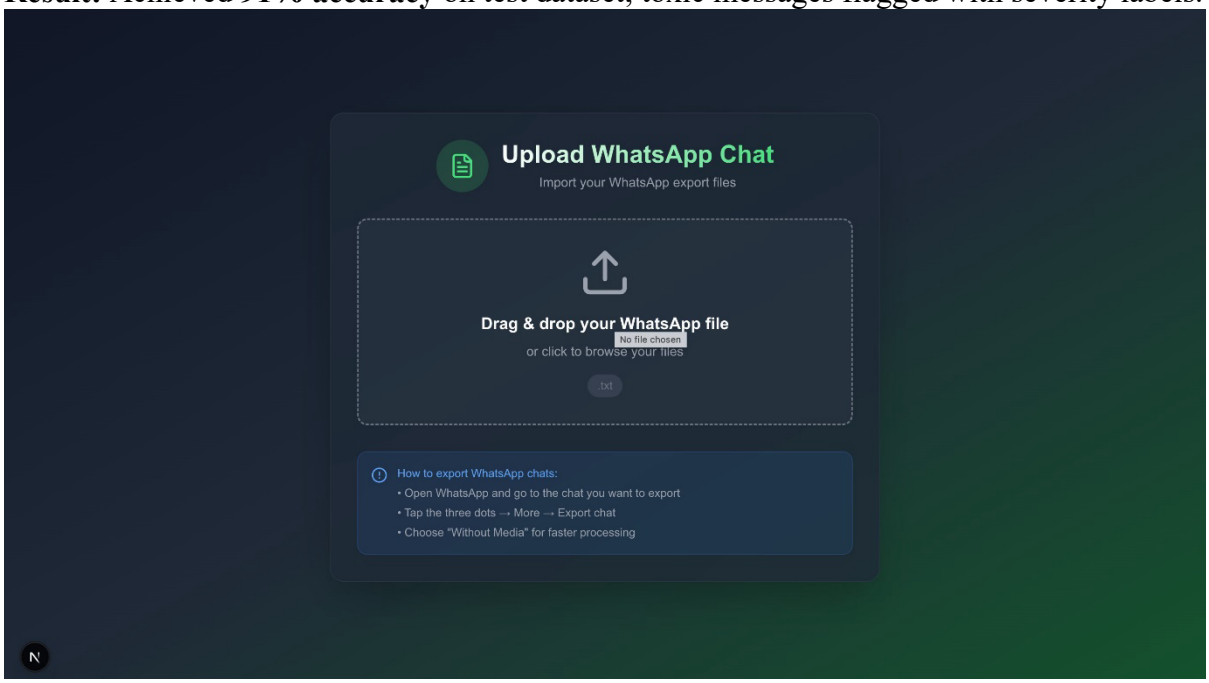

Figure 6: Upload Whatsapp Chat

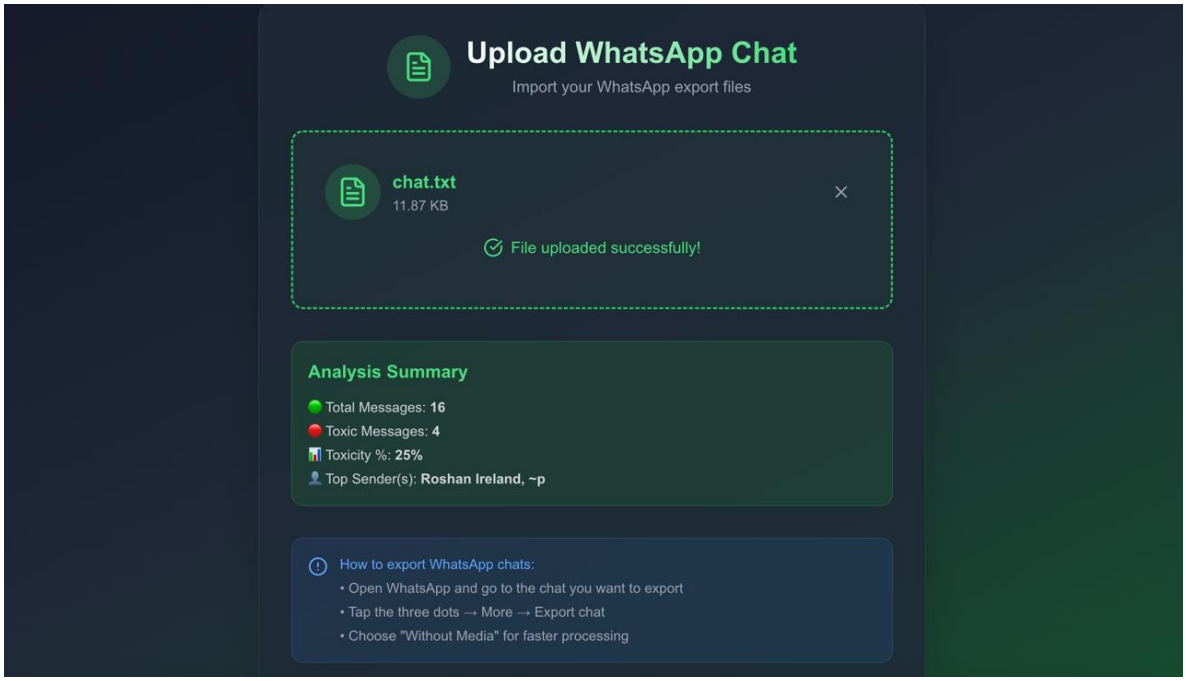

Figure 7: Chat Analysis

6 ALEAPP Parsing Configuration

Setup:

- Configure parsing options (full device timeline, selective app logs, deleted messages).

Enable selection of report format (summary/detailed) (Newman *et al.*, 2019).
Result: Successfully extracted app activity logs, deleted chat events, and suspicious actions for forensic correlation.

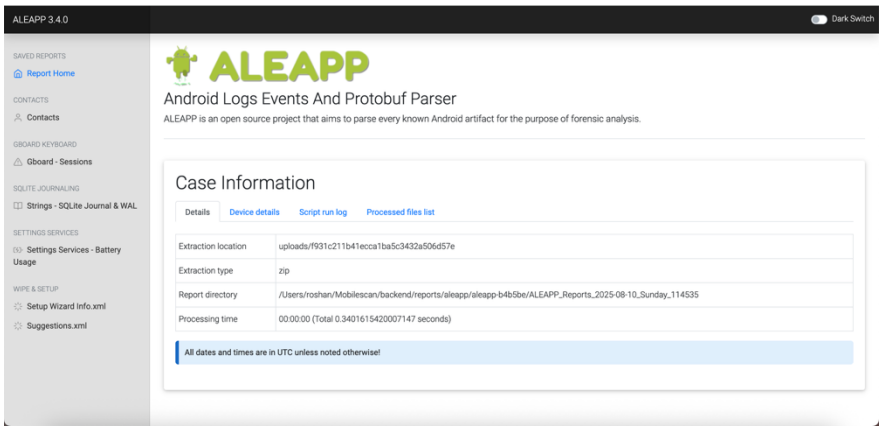

Figure 8: ALEAPP Parser

7 Result Aggregation & Reporting Configuration

Setup:

- Define output formats (PDF, CSV, JSON).
- Configure auto-aggregation of YARA, Regex, RNN, and ALEAPP results.
- Ensure legal admissibility formatting.

Result: Generated structured, timestamped reports ready for court presentation

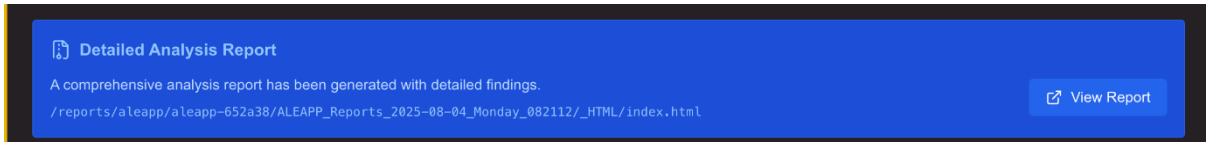

Figure 9: Analysis Report

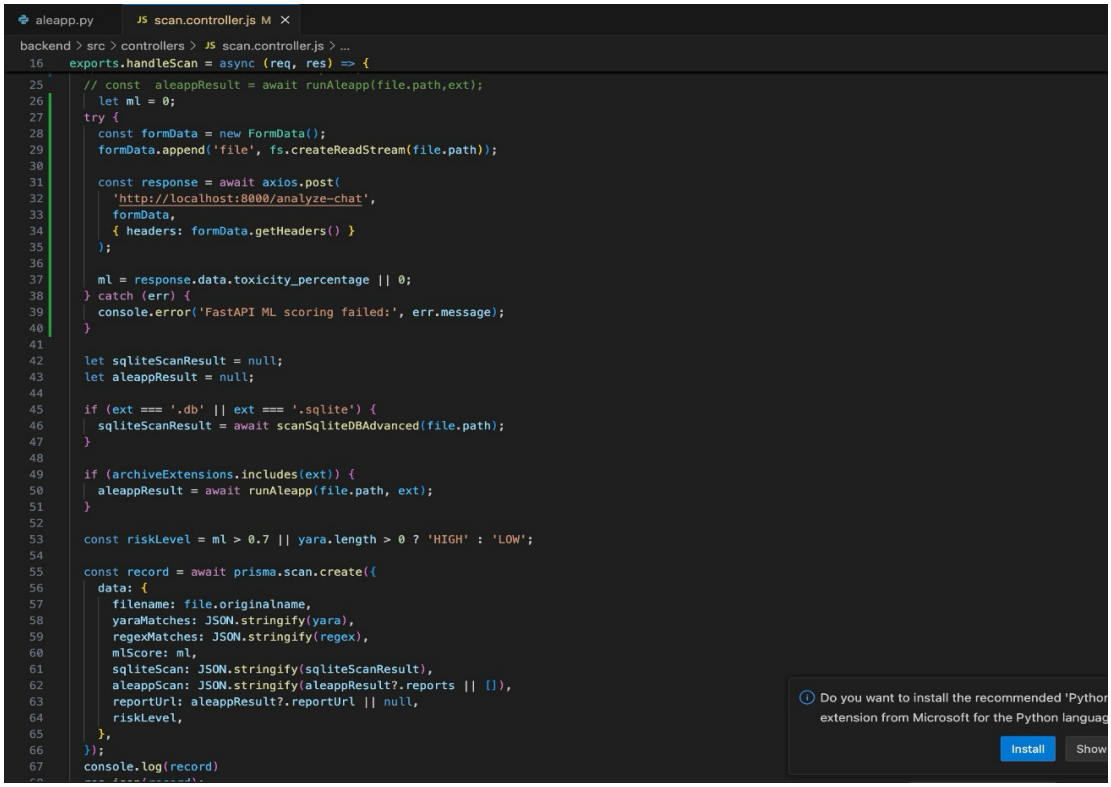

Figure 10: Code for Scanning

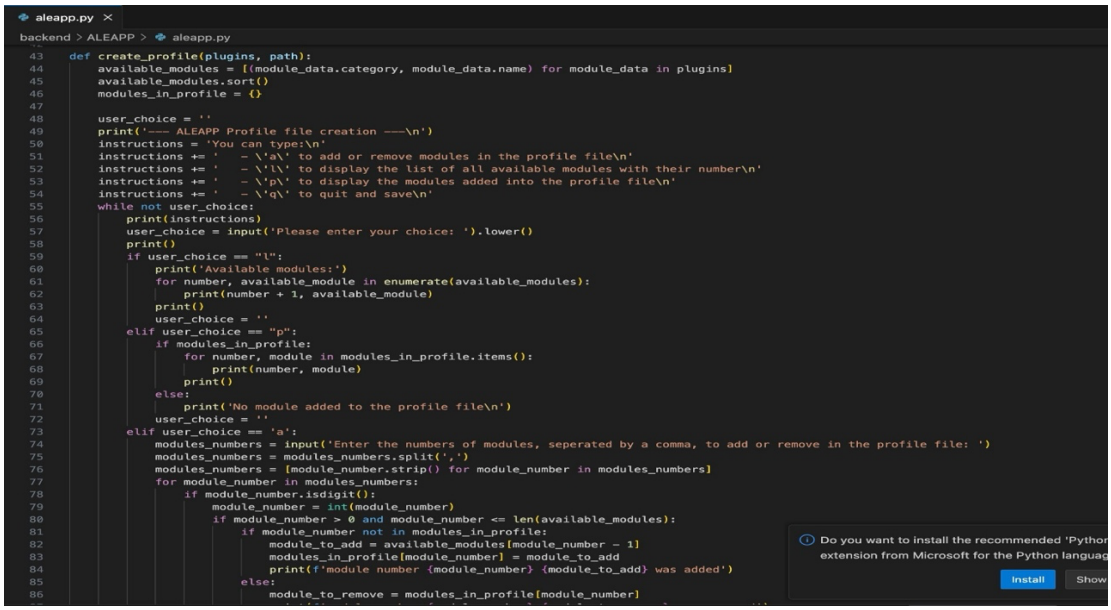

Figure 11: Aleapp Module

7 File-Type Based Scan Examples

This section illustrates the platform's behavior when scanning different evidence formats such as .txt, .zip, and SQLite files. The file types trigger appropriate detection procedures such as YARA signature matching for known malware signatures, regex-based Indicators of Compromise (IoC) extraction for suspicious domains or emails, and heuristic entropy analysis for obfuscated content. The system dynamically picks the scanning type based on the selected

file to ensure precise risk classification and output consistency. Figures labeled File-Type Based Scan provides the capabilities, highlighting how the platform consolidates results into structured PDF and CSV reports. These examples also demonstrate the system's adaptability in processing various digital evidence formats and presenting investigators with clear evidence for further proceedings. This reference will guide you to expected output when scanning different types of files during investigations.

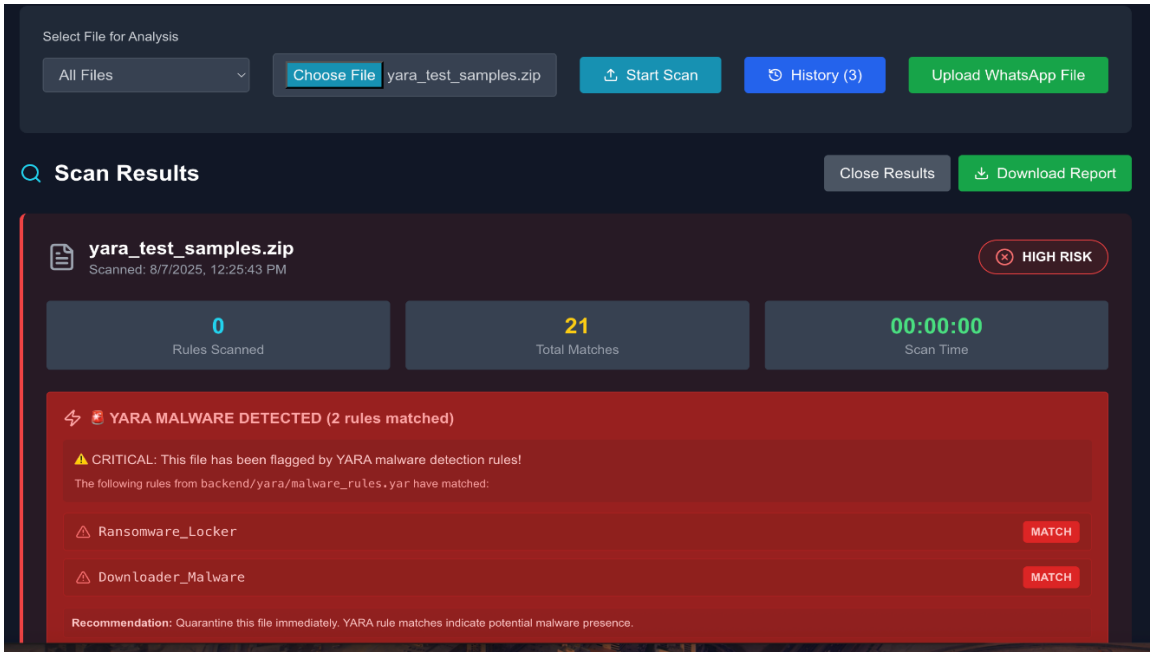

Figure12: ZIP Archive Scan Result (yara_test_samples.zip)

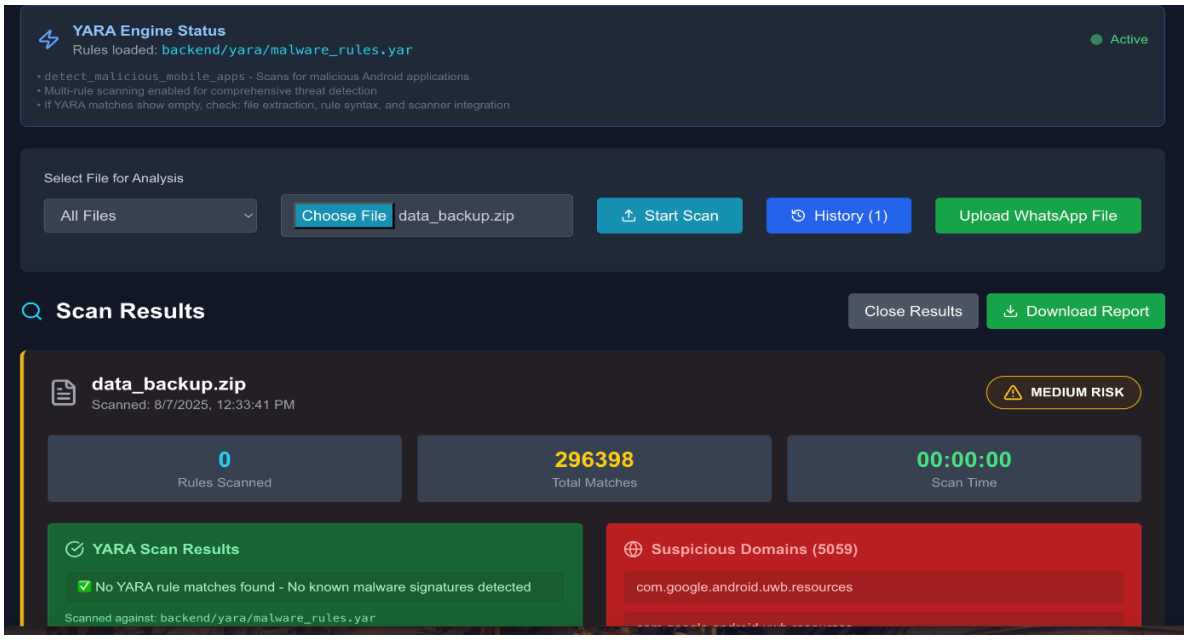

Figure 13: ZIP Archive Scan with Suspicious Domains (data_backup.zip)

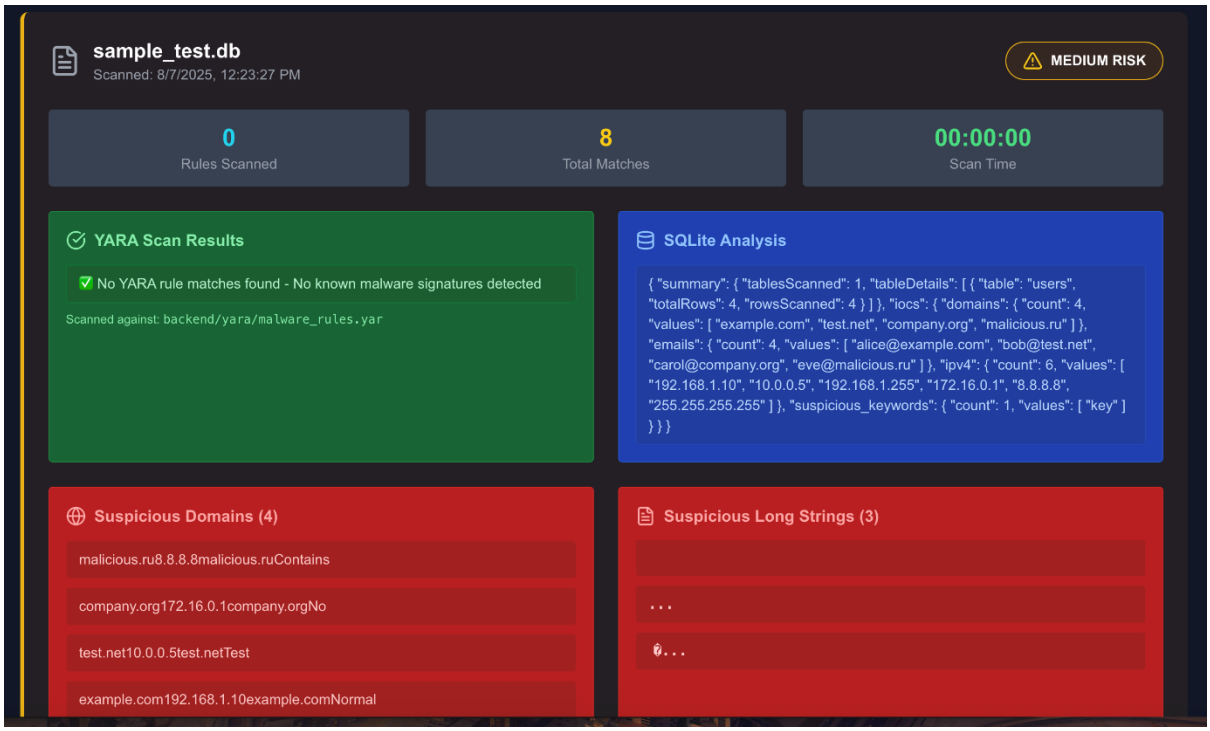

Figure 14. SQLite Database Scan Result (sample_test.db)

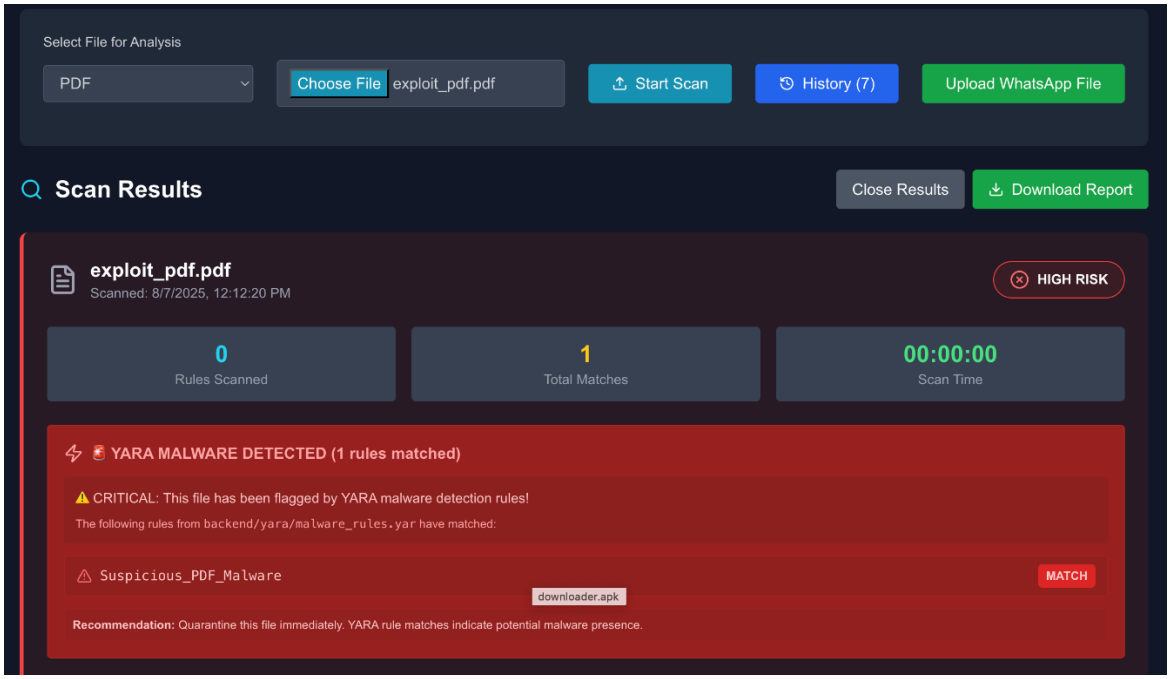

Figure 15: PDF File Scan Result (exploit_pdf.pdf)

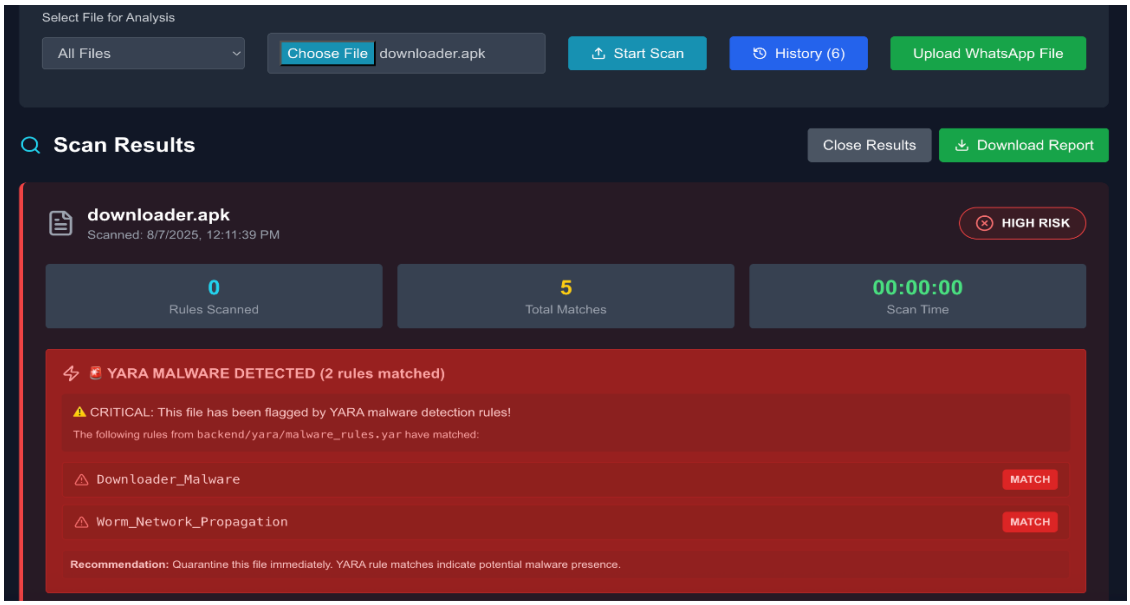

Figure 16: Android APK Scan Result (downloader.apk)

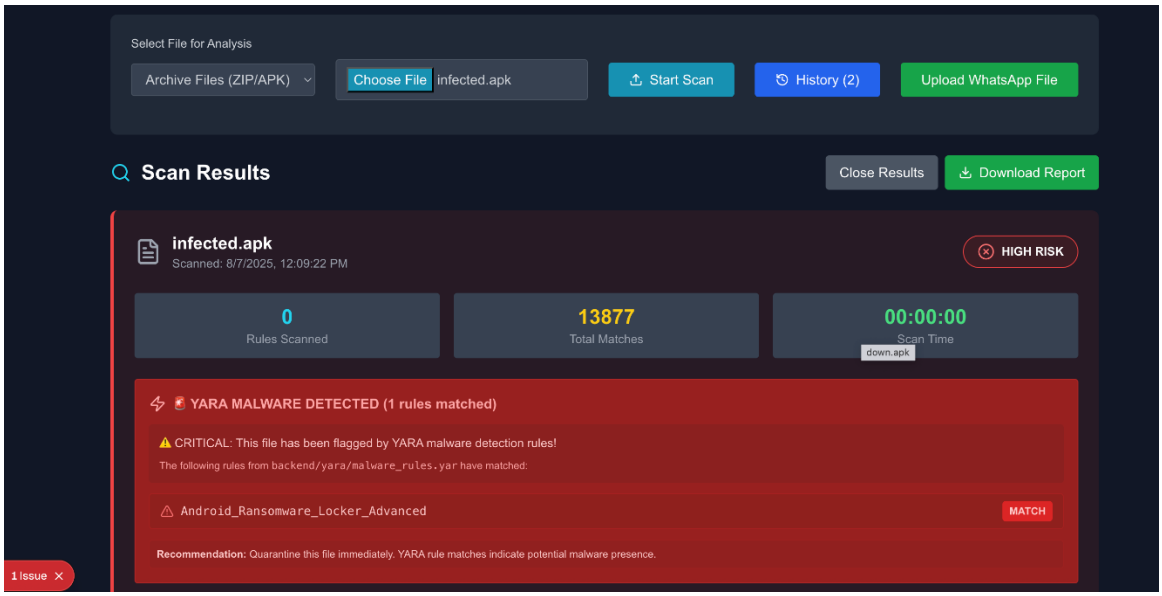

Figure 17: Infected Android APK Scan Result (infected.apk)

8 Android Device Acquisition (ADB Pull)

To extend forensic capabilities beyond file uploads, the system supports the acquisition of Android device data using Android Debug Bridge (ADB). Investigators must first enable Developer Options and USB Debugging on the target device. The Android Developer Tools package is then used to execute commands such as:

(adb pull /data/data /local/evidence/) This command securely copies app databases, logs, and configuration files from the device to a local evidence folder. The acquired data is subsequently processed by ALEAPP, reconstructing device timelines, deleted events, and usage history in detailed reports. This integration ensures analysts can correlate malware detections and chat analyses with device-level activities for a comprehensive case view. Proper chain-of-custody procedures and secure storage of the extracted evidence are essential to maintain evidentiary

integrity. Testing was performed on multiple physical Android devices running Android 9 to Android 13, ensuring compatibility and realistic artifact capture. Physical devices were selected instead of emulators to ensure authenticity of results, as emulator data often lacks real-world behavioral traces. This integration allows investigators to correlate YARA-based malware detections and RNN-based chat toxicity findings with device-level evidence for a complete forensic profile. It is important to follow proper chain-of-custody documentation and maintain secure storage of extracted evidence to preserve admissibility in legal proceedings.

9 Troubleshooting

Common issues include:

- **YARA Rules Not Loading** → Verify rule file paths and syntax.
- **Regex Patterns Not Matching** → Validate regex expressions and test on sample data.
- **RNN Model Misclassification** → Check model weights and threshold settings.
- **ALEAPP Parsing Errors** → Ensure log extraction matches supported formats.
- **Report Not Generating** → Confirm file write permissions and correct export path.

10 Features & Security Review

- PWA Offline-First Forensic Platform
- YARA-Based Static Malware Detection
- Regex Pattern-Based IoC Extraction
- RNN-Based WhatsApp Toxicity Analysis (91% accuracy)
- ALEAPP Timeline and Event Parsing
- Secure Local Data Processing (No Cloud Dependency)

11 Best Practices & Enhancements

- Periodic YARA Rule Updates for Latest Threats
- Regular RNN Model Retraining with New Chat Data
- Automated Backups of Configurations and Scan Results
- Regex Pattern Expansion for Emerging IoCs
- Audit Logging of All Scans for Chain-of-Custody
- Integration Testing Across Browsers for Consistency

References

Brassard-Gourdeau, É. and Khoury, R. (2020) 'Using Sentiment Information for Preemptive Detection of Toxic Comments in Online Conversations'. Available at: https://arxiv.org/abs/2006.10145.

Gupta, S. *et al.* (2024) 'Living off the Analyst: Harvesting Features from Yara Rules for Malware Detection'. Available at: https://arxiv.org/abs/2411.18516.

Mobile Forensic (2023) 'Mobile Forensics – YCSC'. Available at: https://ycsc.org.uk/mobile-forensics/ (Accessed: 6 August 2025).

Newman, J. *et al.* (2019) 'StegoAppDB: a Steganography Apps Forensics Image Database'. Available at: https://arxiv.org/abs/1904.09360.