ITT593 PROJECT DETAILS: PYTHON-BASED DIGITAL FORENSIC TOOL

2. Description

This project involves designing and implementing a digital forensic tool using Python to solve a specific forensic problem. The tool should assist in extracting, analyzing, or reporting digital evidence in scenarios such as file recovery, log analysis, malware analysis, or network traffic investigation.

Students will research existing tools and identify a gap or limitation that their project can address. They will use Python libraries such as Scapy, PyShark, Volatility, pandas, or others relevant to their chosen forensic application.

This project MUST also be incorporated:

- Evidence hashing (SHA256/MD5)

- Chain-of-custody documentation

- Ethical handling of evidence to satisfy CLO3 requirements.

3. Objectives

• Develop a Python-based tool to address a specific digital forensic challenge.

• Understand the principles and methodologies of digital forensics and apply them practically.

• Analyze the forensic data and provide meaningful insights or visualizations.

• Learn to integrate Python libraries and modules to create a robust forensic solution.

• Demonstrate the tool through a practical use case or simulation.

• Apply proper evidence handling, hashing and chain-of-custody procedures (CLO3).

4. Scope

• **Input Data:** Students can use sample forensic datasets such as log files, memory dumps, network captures, or disk images.

• **Output:** The tool should produce clear and actionable forensic insights, such as extracted data, anomaly reports, or visualized results.

• **Applications:** Examples include file carving, IP geolocation for malicious actors, network traffic analysis, malware signature detection, or timeline reconstruction.

• **Evidence Handling:** Must include hashing and chain-of-custody documentation.

5. Deliverables

• **Python Script/Tool:** A functional forensic tool with clear documentation.

• **Project Report:** A comprehensive report covering research, implementation, hashing, CoC, ethics, and results.

• **Presentation and Demo:** Demonstration of tool functionality and forensic findings.

• **Case Simulation:** Show how the tool performs analysis on a dataset.

6. Suggested Features / Ideas

• **File Recovery Tool:** Extract deleted files from disk images using carving techniques.

• **Log Analysis:** Parse and analyze system or server logs to detect suspicious activity.

• **Memory Analysis:** Analyze RAM dumps for processes, network connections, or malware artifacts.

• **Network Traffic Analysis:** Detect anomalies in PCAP files.

• **Email Header Analysis:** Investigate phishing or spoofing.

• **Malware Behavior Analysis:** Detect malicious patterns via signature or hash matching.

7. Requirements

• The tool must be coded in Python.

• Students should use at least two Python libraries relevant to their tool's functionality.

• The tool must handle input data, process it, and produce meaningful outputs.

• Must include evidence hashing + chain-of-custody.

• Emphasis on usability, clarity, and documentation.

8. Evaluation

The project will be assessed based on tool functionality, usability, originality, correct evidence handling procedures, and overall report + presentation quality.

(Full rubric appended at the end of document.)

Examples of Python-Based Digital Forensic Tools

1. **File Carving Tool**

Objective: Recover deleted files from disk images.

Libraries: pytsk3, volatility, pandas, scapy.

Workflow: Load image → detect signatures → reconstruct files.

2. **Log Analysis Tool**

Objective: Detect abnormal log patterns.

Libraries: pandas, pylogrus, matplotlib.

Workflow: Parse logs → identify anomalies → visualize.

3. **Network Traffic Analyzer**

Objective: Identify malicious network patterns.

Libraries: PyShark, Scapy, matplotlib.

4. **Memory Forensics Tool**

Objective: Detect malicious processes.

Libraries: Volatility, pandas, pytsk3.

5. **Email Header Analyzer**

Objective: Detect phishing/spoofing.

Libraries: email, pyzmail, dns.resolver.

Resources for Students

Digital Forensics Python Libraries:

- pytsk3
- Volatility
- Scapy
- PyShark
- pandas
- matplotlib/seaborn

Tutorials:

- Udemy Python for Cybersecurity
- Cybrary Digital Forensics
- SANS Forensic Resources

Datasets:

- NIST CFTT datasets
- Honeynet Project datasets

Tips for Students

• Start small and expand features.

• Use existing tools for inspiration.

• Document everything.

• Simulate real forensic scenarios using sample datasets.

CHAIN OF CUSTODY (CoC) FORM TEMPLATE

Case Title: _____

Group Members: _____

Course & Section: _____

Date Started: _____

Evidence Details:

Evidence Type: _____

File Name: _____

File Size: _____

Evidence Source: _____

Original Hash (SHA256/MD5): _____

Acquisition Date/Time: _____

Handling Log:

No.	Date/Time	Handler Name	Action Taken	Purpose	Hash Verified (Yes/No)
1			Acquisition		
2			Transfer		
3			Analysis Copy Created		
4			Processed Using Tool		
5			Reporting		

Storage Information: -

Storage Location: _____

Backup Location: _____

Access Restriction: _____

Final Hash Verification:

Hash BEFORE Analysis: _____

Hash AFTER Analysis: _____

Integrity Status: MATCHED / NOT MATCHED

Declaration:

I/We declare that the evidence was handled ethically and professionally.

Name	**Signature**	**Date**

STUDENT REPORT TEMPLATE (ITT593)

1. Title Page
• Project title
• Group members
• Course & lecturer
• Date

2. Abstract
Short summary of tool and findings.

3. Introduction
• Background
• Problem statement
• Objective
• Scope

4. Literature Review
• Similar tools
• Gaps identified

5. Methodology
• Tool architecture
• Libraries used
• Workflow

6. Evidence Handling (CLO3)
• Dataset info
• Hashing results
• Chain of Custody form (*refer template given)
• Ethical/legal considerations

7. Implementation
• Screenshot of tool
• Code explanation

8. Case Scenario Simulation
• Dataset
• Steps performed
• Output + screenshots

9. Results & Analysis
Interpret forensic findings.

10. Discussion
Strengths, limitations, improvements.

11. Conclusion

12. References (APA/IEEE)

13. Appendices
• Full code
• Extra screenshots

ITT593 Project Rubric: Python-Based Digital Forensic Tool (20%)

These rubric details the expectations for the Python-Based Digital Forensic Tool project. Your final score will be based on the degree to which your project meets the standards described in the achievement levels below.

1. Tool Functionality & Technical Quality (40%) – 8 Marks

This criterion assesses the technical merit, reliability, and coding standards of the Python tool.

Marks	**8 (Excellent)**	**5–7 (Good)**	**3–4 (Satisfactory)**	**0–2 (Needs Improvement)**
Tool Functionality & Technical Quality	**Exemplary Performance.** The tool runs flawlessly, is highly robust, and implements **more than 3** distinct, complex, and relevant forensic features. It utilizes **more than 2** appropriate Python libraries effectively.	**Strong Performance.** The tool runs reliably with minimal or no minor bugs and implements **at least 3** relevant forensic features. It utilizes **at least 2** appropriate Python libraries.	**Basic Performance.** The tool runs but may contain minor bugs or limited functionality. Implements **fewer than 3** basic forensic features. It may use only one or poorly integrated libraries.	**Poor Performance.** The tool fails to run, contains major errors, or provides minimal/incorrect output. Forensic features are either missing or non-functional.
Code Quality & Output	Code is exceptionally well-structured, highly modular (using functions/classes), and comprehensively commented with docstrings and in-line explanations. The output is professional, easy to interpret, and **100% accurate**.	Code is well-structured, modular, and clearly commented. The output is meaningful, accurate, and easy to follow.	Code is functional but may lack proper structure, modularity, or sufficient commenting. Output is somewhat accurate but may be difficult to interpret.	Code is spaghetti-like, uncommented, or non-functional. Output is confusing, mostly inaccurate, or non-existent.

2. Forensic Process & CLO3 – Evidence Integrity + CoC (20%) – 4 Marks

This criterion assesses adherence to fundamental digital forensic best practices, specifically focusing on CLO3 (Evidence Integrity).

Marks	**4 (Excellent)**	**3 (Good)**	**2 (Satisfactory)**	**0–1 (Needs Improvement)**
Forensic Process & CoC	**Comprehensive & Correct.** The report demonstrates a thorough understanding and application of evidence integrity: * **Correct hashing** (MD5/SHA256) is flawlessly performed by the tool and/or documented. * The **original vs. copy** procedure is clearly, correctly, and ethically explained. * A perfectly **completed Chain-of-Custody (CoC) form** is included and correctly filled.	**Mostly Correct.** Shows a good understanding of evidence integrity: * Correct hashing is performed. * The original vs. copy procedure is explained. * A **completed CoC form** is included with only minor omissions or errors.	**Basic Adherence.** Shows a basic grasp of the process: * Hashing is mentioned but may be inconsistent or incorrectly applied. * The explanation of procedures is weak or missing key steps. * A partially completed or missing CoC form.	**Fails to Meet Standard.** Lacks evidence of adherence to forensic procedure. Hashing is incorrect or not mentioned. No CoC form is included.
Ethical & Legal Considerations	**Insightful & Relevant.** Ethical and legal considerations (e.g., privacy, warrants, relevant legislation) are clearly identified, discussed, and linked directly to the tool's function and scenario.	**Stated & Relevant.** Ethical and legal considerations are clearly stated and relevant to the case scenario.	**Stated but Vague.** Ethical and legal considerations are mentioned but are generic, vague, or lack direct relevance to the project.	**Missing.** Ethical and legal considerations are not mentioned or are irrelevant.

3. Case Scenario & Forensic Analysis (20%) – 4 Marks

This criterion assesses the practical application of the tool and the quality of the derived forensic findings.

Marks	**4 (Excellent)**	**3 (Good)**	**2 (Satisfactory)**	**0–1 (Needs Improvement)**
Case Scenario & Application	**Highly Realistic & Perfectly Applied.** The case scenario is clear, realistic, and complex. The tool is applied flawlessly and appropriately to the provided/created dataset.	**Clear & Correctly Applied.** The case scenario is clear and believable. The tool is applied correctly to the dataset.	**Scenario is Vague/Simple.** The scenario is basic or lacks detail. The tool application is attempted but may have minor procedural errors.	**Scenario is Missing/Irrelevant.** The scenario is non-existent or the tool is incorrectly applied, making the analysis invalid.
Analysis & Findings	**Accurate, Insightful, & Fully Supported.** The interpretation of results is **100% accurate** and provides deep, relevant insights. Findings directly address the scenario and are robustly supported by clear evidence/output from the tool.	**Accurate & Relevant.** Interpretation of results is accurate. Findings are relevant to the scenario and well-supported by the tool's output.	**Mostly Accurate but Limited.** Interpretation is generally accurate but lacks depth. Findings are mentioned but are weakly supported by the tool's output.	**Inaccurate/Missing.** Interpretation is incorrect or absent. Findings are irrelevant, not supported, or lead to an incorrect conclusion.

4. Report Quality (10%) – 2 Marks

This criterion assesses the professionalism and structure of the accompanying project report.

Marks	2 (Excellent)	1.5 (Good)	1 (Satisfactory)	0–0.5 (Needs Improvement)
Report Quality	**Professional & Flawless.** Report is professionally formatted, highly structured, and follows all guidelines (e.g., clear methodology, proper citation). It includes high-quality, clearly labeled screenshots and is free of errors.	**High Quality.** Report is well-formatted and clearly structured. Includes a clear methodology and relevant screenshots. Contains minimal spelling/grammar errors.	**Adequate.** Report is generally structured but may lack professional formatting (e.g., inconsistent headings, poor layout). Methodology is vague or screenshots are poorly labeled. Contains noticeable errors.	**Poor Quality.** Report is poorly formatted, lacks structure, or misses key sections (methodology, evidence). Filled with errors and is difficult to read.

5. Presentation & Demonstration (10%) – 2 Marks

This criterion assesses the team's ability to present their work, demonstrate the tool, and handle Q&A.

Marks	2 (Excellent)	1.5 (Good)	1 (Satisfactory)	0–0.5 (Needs Improvement)
Presentation & Demonstration	**Engaging & Perfect Execution.** The explanation of the tool's function and findings is exceptionally clear and engaging. Teamwork is seamless. All Q&A is handled confidently and	**Clear & Effective.** Explanation is clear. Teamwork is evident. Q&A is handled well, with correct answers. The demonstration is successful and shows the tool's main functions.	**Acceptable.** Explanation is understandable but lacks enthusiasm/detail. Teamwork is weak (e.g., one person dominates). Q&A is attempted but answers are weak. The demonstration is attempted but encounters minor technical issues.	**Poor.** Explanation is confusing or rushed. No evidence of teamwork. Q&A is poorly handled or refused. The demonstration fails or does not showcase the tool's capabilities.

	correctly. The demonstration is flawless and highlights key features.			

TOTAL MARKS: 20%