

IT 360: Final Project Deliverables

Project Overview

This final project requires you to develop a tool or script that addresses a specific problem in the field of digital forensics. You will be responsible for the entire project lifecycle, from conception and development to documentation and presentation. The goal is to apply the concepts learned in class to a practical, hands-on challenge.

Total Points: 100

Submission Guidelines

Your entire project will be submitted as a single link to a private GitHub repository. You must add your instructor as a collaborator to the repository before the deadline. **No other submission methods will be accepted.**

The repository must contain three core components: the programming project, a detailed written report, and a video presentation.

Breakdown of Deliverables

1. GitHub Repository & Programming Project (50 Points)

This is the core of your project. Your repository should be well-organized and contain a functional digital forensics tool.

- **Functionality (30 points):** The tool must work as described and successfully perform a digital forensics task (e.g., metadata extraction, file carving, log analysis, etc.).
- **Code Quality (10 points):** Code should be clean, well-commented, and efficiently written.
- **Repository & README.md (10 points):** The repository must be logically structured. The `README.md` file in the root directory must contain a project overview, a list of features, and clear instructions on how to set up and run your tool.

2. Written Report (30 Points)

A detailed report documenting your project must be included in the repository, preferably in PDF format. The report should follow this structure:

- **Introduction (5 points):** What is the purpose of your project? What problem in digital forensics does it solve?
- **Technical Implementation (10 points):** Describe the technical details of your project. What languages, libraries, and techniques did you use, and why?
- **Results (5 points):** Provide a clear outline of the results your tool produces¹. Include sample outputs, screenshots, or generated reports.
- **Lessons Learned & Conclusion (10 points):** Conclude your report with a section on **lessons learned**². This section should detail **what did and didn't work** during your project development and what you would do differently in the future³.

3. Video Presentation (20 Points)

You will record a 10-15 minute video presentation of your project. Upload the video to a streaming service (like YouTube or Vimeo as an unlisted video) and include the link in your repository's `README.md` file.

- **Content & Clarity (10 points):** The presentation should clearly explain the project's purpose, functionality, and results in a professional manner.
 - **Tool Demonstration (10 points):** You must include a live demonstration of your tool in action, walking through its features and showing the output.
-

Recommended GitHub Repository Structure

To maintain a clean and organized project, please follow this general folder structure within your repository.

```
/your-project-name
|
|-- src/
|   |-- main.py
|   |-- module1.py
|   |-- ... (other source code files)
|
|-- docs/
|   |-- final_report.pdf
|   |-- ... (any other supporting documents or images for the report)
|
|-- data/
|   |-- sample_evidence.dd
|   |-- ... (any sample files your tool needs to run for a demo)
|
|-- .gitignore
|-- LICENSE
|-- README.md      <-- (Must contain project overview, setup instructions, and video link)
|-- requirements.txt <-- (List of Python packages or other dependencies)
```

Grading Rubric

This rubric details how the final project will be evaluated. Each component will be assessed based on the quality and completeness of the work submitted.

Criteria	Exemplary (Full Points)	Proficient (75%)	Developing (50%)	Unsatisfactory (< 50%)	Points
Programming Project (50 pts)					
Functionality	Tool is fully functional, handles errors gracefully, and performs its forensic task effectively and efficiently.	Tool is functional and performs its core task, but may have minor bugs or unhandled edge cases.	Tool has significant bugs, is partially functional, or fails to complete its core task reliably.	Tool is non-functional, does not run, or fails to perform its intended task.	/ 30
Code Quality	Code is exceptionally clean, well-commented, logically structured, and follows best practices.	Code is readable and adequately commented. The structure is logical.	Code is difficult to read, lacks sufficient comments, or is poorly structured.	Code is disorganized, uncommented, and difficult to comprehend.	/ 10
Repository & README.md	Repository is perfectly organized. README.md is professional, comprehensive, and provides flawless setup/run instructions.	Repository is well-organized. README.md contains all required information and instructions are clear.	Repository is disorganized, or the README.md is missing key information or has unclear instructions.	Repository is a mess of files. README.md is missing or completely inadequate.	/ 10
Written Report (30 pts)					
Intro & Technical Details	Purpose is expertly defined. Technical explanation is thorough, insightful, and demonstrates a deep understanding of the concepts.	Purpose is clearly stated. Technical details are accurate and cover all necessary aspects of the implementation.	Purpose is unclear or poorly defined. Technical details are superficial or contain inaccuracies.	Purpose is not defined. Technical details are missing or fundamentally incorrect.	/ 15

Criteria	Exemplary (Full Points)	Proficient (75%)	Developing (50%)	Unsatisfactory (< 50%)	Points
Results & Lessons Learned	Results are clearly presented and analyzed ¹ . Lessons learned section offers deep, insightful reflection on the project's challenges and successes ² .	Results are presented clearly ³ . Lessons learned section effectively discusses what worked and what didn't ⁴ .	Results are confusing or incomplete ⁵ . Lessons learned section is superficial or simply lists problems without reflection ⁶ .	Results are missing or incomprehensible ⁷ . Lessons learned section is absent ⁸ .	/ 15
Video Presentation (20 pts)					
Content & Clarity	Presentation is engaging, professional, and explains the project with exceptional clarity. The 10-15 minute time limit is used effectively.	Presentation is clear, covers all required points, and adheres to the time limit.	Presentation is difficult to follow, misses key points, or is significantly over/under the time limit.	Presentation is unclear, unprofessional, and fails to explain the project.	/ 10
Tool Demo	The demonstration is smooth, comprehensive, and effectively showcases the tool's full capabilities and outputs.	The demonstration is successful and clearly shows the tool's core functionality.	The demonstration is flawed, fails to show key features, or the tool crashes.	No demonstration is provided, or the tool does not work during the demo.	/ 10
				Total	/ 100