Internship Assessment Document

Title: Tamper Detection in Academic Credentials

Role: AI Development Intern

Submission: GitHub + Google Drive (or any file share link)

🔍 Background

Educational document fraud is a growing problem, with nearly 15–20% of academic credentials containing falsifications. Sophisticated digital tools have made it easier to alter PDF metadata, forge seals, and manipulate grades or certification details.

As a member of the Credential Verification Research Team, you are expected to explore the technical feasibility of automated tamper detection in educational documents like:

- Degree Certificates
- Academic Transcripts
- Professional Certifications

🎯 Objective

Your mission is to design and prototype a tamper detection system that flags manipulated academic documents based on:

- PDF structure and metadata
- Visual content and layout irregularities
- Document type–specific tampering patterns

🧪 Assessment Tasks

- Task 1: Document Analysis

Identify and document common tampering patterns in:

- Degree Certificates (PDF metadata changes, fake templates)
- Academic Transcripts (grade edits, course list changes)
- Professional Certifications (date extensions, level changes)

- Task 2: Prototype Development

Develop a lightweight script or prototype that demonstrates at least one of the following:

- PDF Metadata Analysis – Detect discrepancies like issue date mismatch, edited metadata, etc.
- Template Matching or Layout Analysis – Use visual methods to compare the layout with known valid templates
- Anomaly Detection – Identify content or format irregularities across multiple samples

 - (Bonus) Image-based Detection – OCR-based text comparison or detection of edited images

· Task 3: Technical Report

Prepare a short report (max 2 pages) covering:
 - Your approach, assumptions, and methodology
 - Challenges and trade-offs
 - Suggestions to improve or scale the system

· Task 4: Demo (Optional but Strongly Recommended)

Record a brief walkthrough (max 3 minutes) of your prototype or key logic and share the video link.

📦 Deliverables

- Source Code (GitHub or zip)

- Technical Report (PDF or Google Doc)

- (Optional) Demo Video Link

- Any mock/sample documents you used or generated

📊 Evaluation Criteria

Category	Weight
Technical Soundness	30%
Understanding of the Problem	20%
Creativity in Approach	20%
Code Quality & Documentation	15%
Communication & Demo	15%

🔧 Tools & Resources

You may use any of the following:
 - Python, JS, or any scripting language

- Pypdf2 / pdfminer / pdfplumber
- Tesseract OCR, OpenCV (for image manipulation)
- Public datasets or self-created mock documents

📬 Submission Instructions

1. Upload your code and report to a GitHub repository or Google Drive folder.
2. Share the link with appropriate access permissions.
3. Deadline: 11th May - 2025