Detailed Project Plan

Phase 1: Planning & Infrastructure Setup (Week 1-2)

1. Requirements Gathering

- Define types of PII to detect (government IDs, emails, phone numbers, addresses, etc.).
- Identify all sources of data (S3 buckets, local storage).
- Compliance requirements (GDPR, HIPAA).
- Look into the requirements of Presidio and get familiar with its working and architecture.

2. Tool & Framework Setup

- Install dependencies such as Python libraries (`tesseract-ocr`, `spacy`, `nltk`, `opencv`, etc.).
- Set up local and cloud infrastructure (ensure access to AWS S3, other cloud storages from Azure, Google and local storage).
- Set up the requirements for Presidio
- Install additional components:
    - Install `tesseract-ocr` for OCR.
    - Install `Spacy` for language processing.
    - Set up OpenCV for image handling and face detection.

3. Architecture Design

- Design data pipeline:
    - Files retrieved from local directories or S3 buckets.
- OCR processing for images.
- NLP and regex-based classification with the help of Presidio.

Phase 2: Data Collection & Preprocessing (Week 3)

4. Data Input Handling

- Write scripts to fetch images, PDFs, and documents from:
- Local filesystems.
- S3 buckets.

5. Preprocessing Images

- Use OpenCV to convert images to grayscale and apply techniques such as auto-rotation, deskewing, thresholding.
- PDF files should be converted into image lists for further processing.
- Preprocess text-based files (txt, csv, doc) by reading and normalizing their content.

6. Face Detection (for images)

- Implement face detection using Haar cascades in OpenCV.
- Filter files that contain faces for further investigation.

Phase 3: PII Detection & Model Training (Week 4-5)

7. OCR & Text Extraction

- Run Tesseract OCR on cleaned images and scanned PDFs.
- Extract text into a structured list format (tokens, strings).
- Perform text normalization (removal of noise, special characters).

8. Regex for PII Identification:

- Implement regular expressions to search for emails, phone numbers, and government ID formats within the extracted text.
- Create custom regex patterns for local types of PII (PAN for India, SSNs for the US, etc.).
- Take the help of Presidio's Analyzer

9. NLP for Advanced PII Detection

- Use Spacy/NLTK to extract contextual information such as addresses and location data.
- Apply named entity recognition (NER) to classify and tag PII types.
- Implement with Presidio's Analyzer

10. Confidence Scoring & PII Classification

- For each detected PII, assign a confidence score based on the number of matches in the predefined keyword lists (using pattern matching).
- Research and find the best confidence scoring parameters for the PII classified data.
- Map files to their respective PII class (e.g., driver's license, passport, etc.) based on repeated matches.

Phase 4: Automation & Integration (Week 6)

11. Automated Scanning

- Set up automation for periodic scanning:
- Use AWS Lambda triggers for S3 bucket updates.
- Local file system changes can be detected via cron jobs.

12. Error Handling & Resilience

- Ensure that OCR can handle low-quality or corrupted images gracefully.
- Implement logging to track failed attempts or unclassified files for later review.

Phase 5: Output Generation & Reporting (Week 7)

13. File Processing & Result Storage

- Write outputs into a `output.txt` file containing:

- File path.
- Detected PII (emails, phone numbers, addresses).
- Country of origin and PII class.
- Face detection results (if any).
- Store results in an organized format (JSON, CSV) for further analysis or compliance checks.

14. Reporting

- Generate reports for stakeholders detailing the detected sensitive information.
- Summarize the quantity and types of PII found per file/source.
- Add features like email notifications or a dashboard for real-time monitoring of results.

Phase 6: Testing, Documentation & Deployment (Week 8)

15. Testing

- Conduct thorough testing with dummy data (e.g., sample PII files).
- Validate accuracy using various file formats (JPG, PNG, PDF, DOC, TXT).
- Test for false positives/negatives, and refine regex/NLP logic accordingly.

16. Documentation

- Document the installation, execution, and troubleshooting steps
- Prepare user manuals and technical documentation for future developers.

17. Deployment

- Deploy the final tool in the production environment, ensuring necessary permissions and security protocols are in place.
- Set up monitoring to log and alert about potential issues or new PII leaks.