📘 Financial Fraud Analyzer – User & Developer Documentation

Industry: Financial Services **Version:** 1.0 **Date:** May 2025

HG **by Hades Gaming**

Made with GAMMA

1. Overview

Financial Fraud Analyzer is an AI-driven tool for automating fraud detection in financial documents such as loan applications and invoices. It combines document understanding (via NLP) with anomaly detection, saving up to 70% of manual review time and increasing fraud flagging accuracy.

Key Capabilities:

- Intelligent parsing of scanned/structured documents (PDFs)
- Anomaly detection using Isolation Forest
- Explainability using SHAP or LIME
- Simple user interface for uploads, results, and exports
- Optional integration with Slack or APIs

Made with GAMMA

2. User Guide

2.1 System Requirements

Requirement	Specification
OS	Windows, macOS, or Linux
Python	3.7 or higher
Browser	Chrome/Firefox (for UI)
RAM	8 GB+ recommended

2.2 Installation & Setup

1 Clone the Repository:

```
git clone https://github.com/Shivasairam1706/Financial-fraud-analyzer.git
cd Financial-fraud-analyzer
```

2 Create Virtual Environment:

```
python -m venv venv
source venv/bin/activate # Windows: venv\Scripts\activate
```

3 Install Dependencies:

```
pip install -r requirements.txt
```

4 Launch the App:

```
streamlit run app.py
```

Made with GAMMA

2.3 Using the Application

Launch Web Interface:

Open browser at http://localhost:8501

Upload Documents:

Select multiple PDFs or images

Run Analysis:

Click "Analyze" to extract and process

View Results:

Output includes:

- Extracted fields (amounts, vendors, etc.)
- Fraud probability score
- Highlighted anomalies

Download Report:

Option to export as CSV or PDF

Made with GAMMA

2.4 Understanding the Output

Field	Description
doc_id	Unique document ID
amount	Total invoice/loan amount
vendor_name	Entity requesting payment
anomaly_score	Outlier score (0–1 scale)
flagged	True if score exceeds threshold
explanation	SHAP/LIME reasoning summary

2.5 Troubleshooting

Issue	Solution
App won't launch	Check Python version, re-install packages
PDF not parsed	Ensure file isn't encrypted or scanned without OCR
No anomalies flagged	Try adjusting anomaly threshold in anomaly_detector.py

Made with GAMMA

3. Developer Guide

3.1 Project Structure

```
Financial-fraud-analyzer/
├──── app.py # Main Streamlit app
├──── parser.py # Document parsing logic (LayoutLM, etc.)
├──── anomaly_detector.py # Isolation Forest implementation
├──── explainer.py # SHAP/LIME logic
├──── models/ # Trained models
├──── utils.py # Common utilities
├──── requirements.txt # Package dependencies
```

3.2 Technology Stack

Component	Technology
Document Parsing	HuggingFace LayoutLM / Donut
Anomaly Detection	Scikit-learn (Isolation Forest)
Explainability	SHAP, LIME
UI	Streamlit
Integration	Slack API (optional)

Made with GAMMA

3.3 Model Training Pipeline

Data Collection:

Use sample financial datasets (from Kaggle or synthetically generated) to build training corpus.

Field Extraction (parser.py):

- Fine-tune LayoutLM/Donut on labeled examples
- Output: structured JSON of key-value pairs

Anomaly Detection (anomaly_detector.py):

- Train Isolation Forest on clean data
- Input: extracted features (amount, vendor frequency, etc.)
- Output: anomaly score (0–1), threshold to flag as fraud

3.4 Explainability Layer

Implemented via SHAP:

- Visual and textual explanations of fraud flags
- E.g., “Invoice amount exceeds vendor's 3-month average”

```
import shap
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X)
```

Alternative: Use LIME for local model interpretability.

Made with GAMMA

3.5 Integration & Deployment

Streamlit UI – Run locally or deploy via:

- **Docker:** Create Dockerfile and expose port 8501
- **Heroku/Render:** Use Procfile and requirements.txt
- **Slack Bot (Optional):** Connect via Slack Events API to alert when fraud is detected

4. Customization & Extensions

Feature	Extension Ideas
Gen AI Layer	Use GPT-4 to draft emails ("Your loan is rejected due to inconsistent income history...")
RAG	Compare documents with internal policy KB
Feedback Loop	Add reviewer override → retrain model
Visual Charts	Add anomaly trends over time (Streamlit Charts)
Batch Processing	Handle zipped documents or folders

Made with GAMMA