AI Insurance Claims Assistant: Automated Claim Analysis

Streamlit App for Image-Based Insurance Reports

Created and Presented by Chekka Vigneshwar

8 September 2025

Introduction

- **Purpose**: Automate insurance claim analysis using AI to compare before and after incident images.
- **Context**: Streamlines claims processing, detects fraud, and generates clean reports.
- **Technology**: Powered by Google Gemini 1.5 Flash and Streamlit.
- **Key Features**:
  - Analyzes before/after images and user descriptions.
  - Detects items, damage, and fraud risk.
  - Exports reports in TXT and DOCX formats.
  - Modern UI with Roboto font and blue accents.

Key Features

- **Image Analysis**:
  - Upload before/after images (jpg, jpeg, png).
  - Compares items and detects damage.
- **Fraud Detection**:
  - Assesses story consistency and fraud risk.
  - Highlights high fraud risk with warnings.
- **Report Generation**:
  - Outputs structured JSON for analysis.
  - Exports clean TXT and DOCX reports.
- **User-Friendly UI**:
  - Roboto font, blue accents (`#0066cc`).
  - Styled warnings for fraud detection.

How It Works

1. **Input Data**:
   - Upload before and after incident images.
   - Provide a text description of the incident.
2. **AI Analysis**: Gemini 1.5 Flash processes images and description to generate:
   - Claim type, severity, and fraud risk.
   - Lists of items (before, after, missing, damaged).
   - Damage summary and recommendation.
3. **Output**: Structured JSON response displayed in a styled UI.
4. **Export**: Downloadable TXT or DOCX report with sanitized text.

Technical Details

- **AI Model**: Google Gemini 1.5 Flash
  - Multimodal (text + images), fast, cost-effective.
  - Generates structured JSON output.
- **Framework**: Streamlit (Python-based web app).
- **Dependencies**:
  - `streamlit`, `google-generativeai`, `pillow`, `python-docx`.
- **Styling**: Custom CSS with Roboto font, blue accents (`#0066cc`), and fraud warnings (`#fff3cd`).
- **Output**: JSON, TXT, and DOCX reports with sanitized text.
- **Error Handling**: Robust handling for API, image, and parsing issues.

Usage Demo

Input Example:

- Description: ”Car hit a pole last night.”
- Images: Before (undamaged car), After (dented bumper).

Output Example (JSON): {

"claim$_t$*ype” : ”VehicleDamage”*,

"severity": "Medium",

"items$_b$*efore”* :

[*”bumper”*, *”headlight”*],

"items$_a$*fter”* :

[*”bumper*(*dented*)*”*, *”headlight”*],

"missing$_i$*tems”* : [],

"damaged$_i$*tems”* : [*”bumper”*],

"damage$_s$*ummary”* :

”Dentonfrontbumper.”,

"consistency$_w$*ith$_s$tory”* : *”Yes”*,

"fraud$_r$*isk”* : *”Low”*,

"recommendation": "Approve claim for bumper repair."

}

Report Example (TXT): Type of Claim: Vehicle Damage

Severity: Medium

Fraud Risk: Low

Recommendation: Approve claim for bumper repair.

Benefits

- **Efficiency**: Automates claim analysis, reducing manual effort.
- **Accuracy**: Compares before/after images for precise damage detection.
- **Fraud Detection**: Identifies inconsistencies and high fraud risk.
- **User-Friendly**: Modern UI with clear section headers and warnings.
- **Flexibility**: Supports TXT and DOCX report exports.
- **Cost-Effective**: Uses Gemini 1.5 Flash for fast processing.

Conclusion

- **Summary**: The AI Insurance Claims Assistant streamlines claim processing with AI-driven image analysis, fraud detection, and clean report exports.
- **Value**: Saves time, improves accuracy, and enhances fraud prevention.