

Code Documentation for Smart Receipt Assistant

Overview

The **Smart Receipt Assistant** is a **Streamlit-based chatbot** that extracts data from receipt images using **OpenAI's GPT-4 Turbo Vision API** and enables users to ask questions about the extracted information.

Features

- **Receipt Data Extraction:** Extracts key details (restaurant name, address, date, itemized purchases, total, tax, payment method) using AI.
- **Chatbot Interaction:** Users can ask questions about their receipt details in a conversational format.
- **Session Management:** Maintains receipt data and chat history for an interactive experience.

Dependencies

- **Python Libraries:** `streamlit`, `openai`, `PIL`, `re`, `base64`, `io`
- **OpenAI API:** Used for image-to-text extraction and chatbot interactions.

Code Breakdown

1. Importing Required Libraries

The script begins by importing the necessary libraries for:

- Streamlit UI (`streamlit`)
- OpenAI API calls (`openai`)
- Image processing (`PIL.Image`, `io`, `base64`)
- Text processing (`re`)

2. ReceiptChatbot Class

This class handles receipt extraction and chatbot interactions.

`__init__` Method

- Initializes `receipt_data` (extracted text) and `image` (uploaded image).

clean_text(text)

- Cleans extracted text by:
 - Removing asterisks (*)
 - Normalizing spaces and newlines

extract_receipt_data(image)

- Converts image to **base64** and sends it to **OpenAI's Vision API**.
- Extracts:
 - Restaurant Name
 - Address
 - Date & Time
 - Itemized List
 - Total Amount & Tax
 - Payment Method (if available)
- Returns cleaned receipt details.

chat_with_receipt(query, chat_history)

- Uses OpenAI's ChatGPT API to **answer questions** about extracted receipt details.
- Maintains **chat history** for context.
- Returns chatbot responses.

3. Streamlit UI (main() function)

st.file_uploader()

- Accepts receipt image uploads (JPG, PNG, JPEG).
- Displays uploaded image.

Receipt Data Extraction

- Automatically extracts and displays receipt details.

Chat Interface

- Users can **ask questions** about the receipt.

- Maintains **chat history**.
- Displays both **user queries** and **chatbot responses**.

Usage

1. Run the script in **Streamlit** (`streamlit run script.py`).
2. Upload a **receipt image**.
3. View extracted details.
4. Ask **questions** about the receipt.

Error Handling

- **Invalid API key:** Shows an error message.
- **No receipt uploaded:** Prevents chat functionality.
- **Exception Handling:** Captures errors in API responses.