# Team Name: AIMarkTech

# Table of Contents • Project Overview • Key Features and Functionalities     ◦ AI-Powered Loan Servicing Request Classification     ◦ Confidence Scoring for AI Predictions     ◦ Duplicate Email Detection     ◦ Context-Based Data Extraction     ◦ Handling Multi-Request with Primary Intent Detection     ◦ Priority-Based Extraction

# • How to Run the Project • Testing and Results

# **• Demo**

# Project Overview

This project leverages generative AI, specifically Google Gemini as the LLM model, to analyze emails and their attachments. It classifies the loan servicing type and sub-request type while providing a confidence score for each prediction. Built using Python, the system automates the classification of service requests, improving efficiency and accuracy in loan servicing operations.

Team Name: AIMarkTech

Git Repo: https://github.com/ewfx/gaied-ai-mark-tech/

# Key Features and Functionalities of the Project

## AI-Powered Loan Servicing Request and Sub request Classification

* + Reads **emails (EML format)** and extracts both the **email body** and **attachments** (PDF, DOCX) and classifies loan servicing requests. Tools supports following file format as input: .eml, pdf and .docx
  + Utilizes **Google Gemini (LLM)** to determine the **loan servicing request type and sub-request type**.
  + Identify the sender intent for email sending.
  + Defines several **loan servicing types** such as:
    - Adjustment
    - AU Transfer
    - Closing Notes
    - Commitment Change
    - Fee Payment
    - Money Movement (Inbound/Outbound)

Configuration Table for Request type and sub request types:

|  |  |
| --- | --- |
| **Request Type** | **Sub Request Type** |
| Adjustment |  |
| AU Transfer |  |
| Closing Notes | Reallocation Fees,Amendment Fees,Reallocation Principal |
| Commitment Change | Cashless Roll, Decrease,Increase |
| Fee Payment | Ongoing Fee, Letter of Credit Fee |
| Money Movement Inbound | Principal, Interest,Principal and Interest,Principal and Interest and Fees |
| Money Movement Outbound | Time bound,Foreign Currency |

* + Uses **detailed instructions and examples** to guide the AI model for **accurate classification**.

## Confidence Scoring for AI Predictions

* + Each classification includes a **confidence score (0.0 - 1.0)** to indicate AI certainty.
  + Helps in determining whether human intervention is needed for low-confidence classifications.

## Duplicate Email Detection

* + Uses **MD5 hashing** to track processed emails and prevent duplicate processing.
  + If an email has already been processed, it returns a duplicate flag.
  + If an email is forwarded in same thread or Email is replied multiple time in same thread then, it provides duplicate flag.
  + Duplicate reason provides the reason for duplication along with flag

1. **Context Based data Extraction**
   * Context based data extraction from email and attachment based on configured fields.
   * Extracting value of Deal, Amount and Expiration date from email and attachment. These fields are configured fields and it read the value from email and attachment. It first read the value from email and post that it look value from attachments. This fields are configurable. We can change the list of fields based on the requirement.
2. **Handling multi request with primary intent detection:**

* + Project supports complex cases where single email can contain multiple request types and It creates multiple request type. Project supports multiple request type identification in single email. It identify the intent ask of sender and identify the request type.
  + Primary request identification which represent sender’s main intent even when the email discusses the multiple topic.

1. **Priority based Extraction**
   * Project supports customization rule such as prioritizing emails content over documents.
   * It supports extracting value from email content or attachments. Configuration is provided to extract value at fields level through email content or Attachments.

# How to Run the Project

## Prerequisites

Before running the project, ensure that you have the following installed:

* + **Python 3.8+**
  + **pip** (Python package manager)
  + **Virtual environment** (Recommended but optional)
  + **Google Gemini API Key**
  + **Download pdf reader, document reader and eml reader library**

## Clone the Repository

Download or clone the project to your local system using:

git clone https://github.com/ewfx/gaied-ai-mark-tech cd gaied-ai-mark-tech

## Set Up a Virtual Environment (Optional but Recommended)

Creating a virtual environment ensures dependencies do not interfere with system-wide Python packages.

python -m venv venv

source venv/bin/activate # On macOS/Linux venv\Scripts\activate # On Windows

## Install Required Dependencie

Run the following command to install all necessary Python packages: pip install openai google-generativeai flask python-docx pymupdf

## Set Up Google Gemini API Key

Ensure you have a valid API key from Google Gemini. Replace Google\_API\_Key in the script or set it as an environment variable:

export GOOGLE\_API\_KEY="your-api-key" # On macOS/Linux set GOOGLE\_API\_KEY="your-api-key" # On Windows

Modify the script to read from the environment variable:

import google.generativeai as genai import os

genai.configure(api\_key=os.getenv("GOOGLE\_API\_KEY"))

## Start the Flask Server

Run the following command to start the server:

python your\_script.py

## Test the API

You can test the /process-file API using **Postman, cURL, or a Python script**.

**Postman Desktop App:**

**End Point URL**: <http://127.0.0.1:5000/process-file>

Input: pdf, .eml and .doc file

## Using cURL:

curl -X POST -F "file=@sample\_email.eml" <http://127.0.0.1:5000/process-file>

## Expected API Response

{

"duplicate\_flag": "No",

"duplicate\_reason": "Unique email.",

"extracted\_fields": {

"Amount": "$1,500",

"Deal": "23333333",

"Expiration Date": "23 October 2025"

},

"requests": [

{

"confidence\_score": 0.95,

"request\_type": "Money Movement Inbound",

"sub\_request\_type": "Principal"

}

]

}

**Testing and Results:**

Test:

Step-1 Create a sample email and provide as input to code:

We have used below attache email for testing.

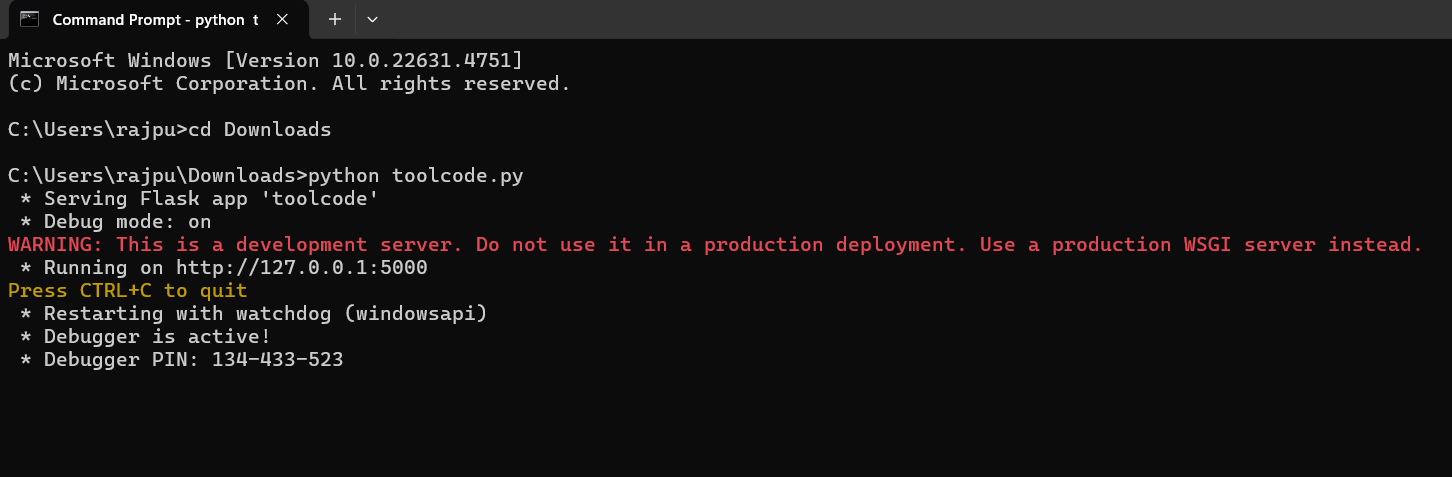


Step-2

Run the attached python code:

Toolcode.py





Step-3

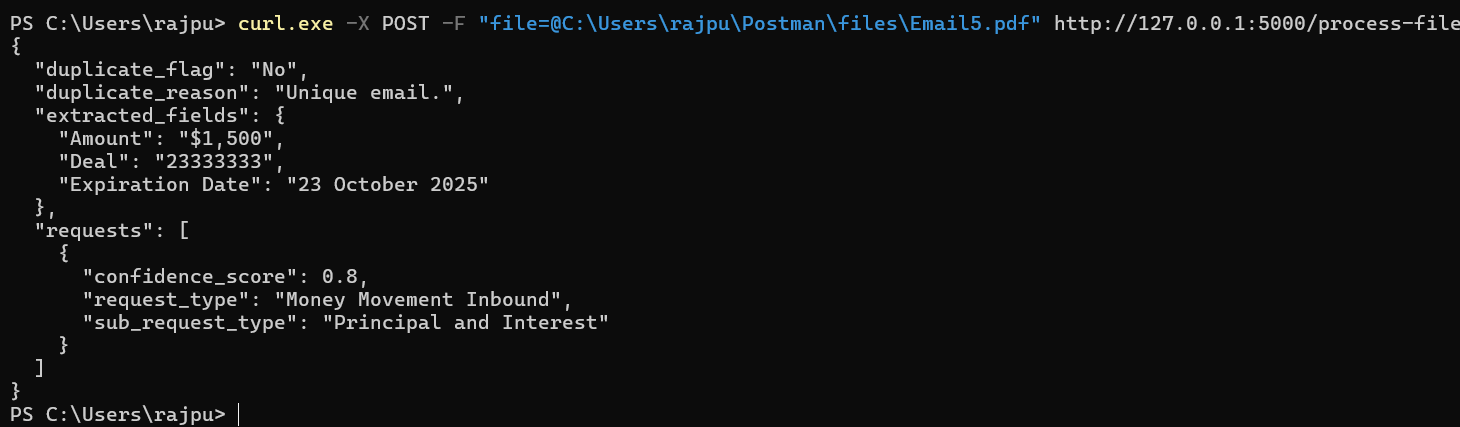
Run the code in cmd or through postman API:

CMD: curl.exe -X POST -F "file=@C:\Users\rajpu\Postman\files\Email5.pdf" <http://127.0.0.1:5000/process-file>



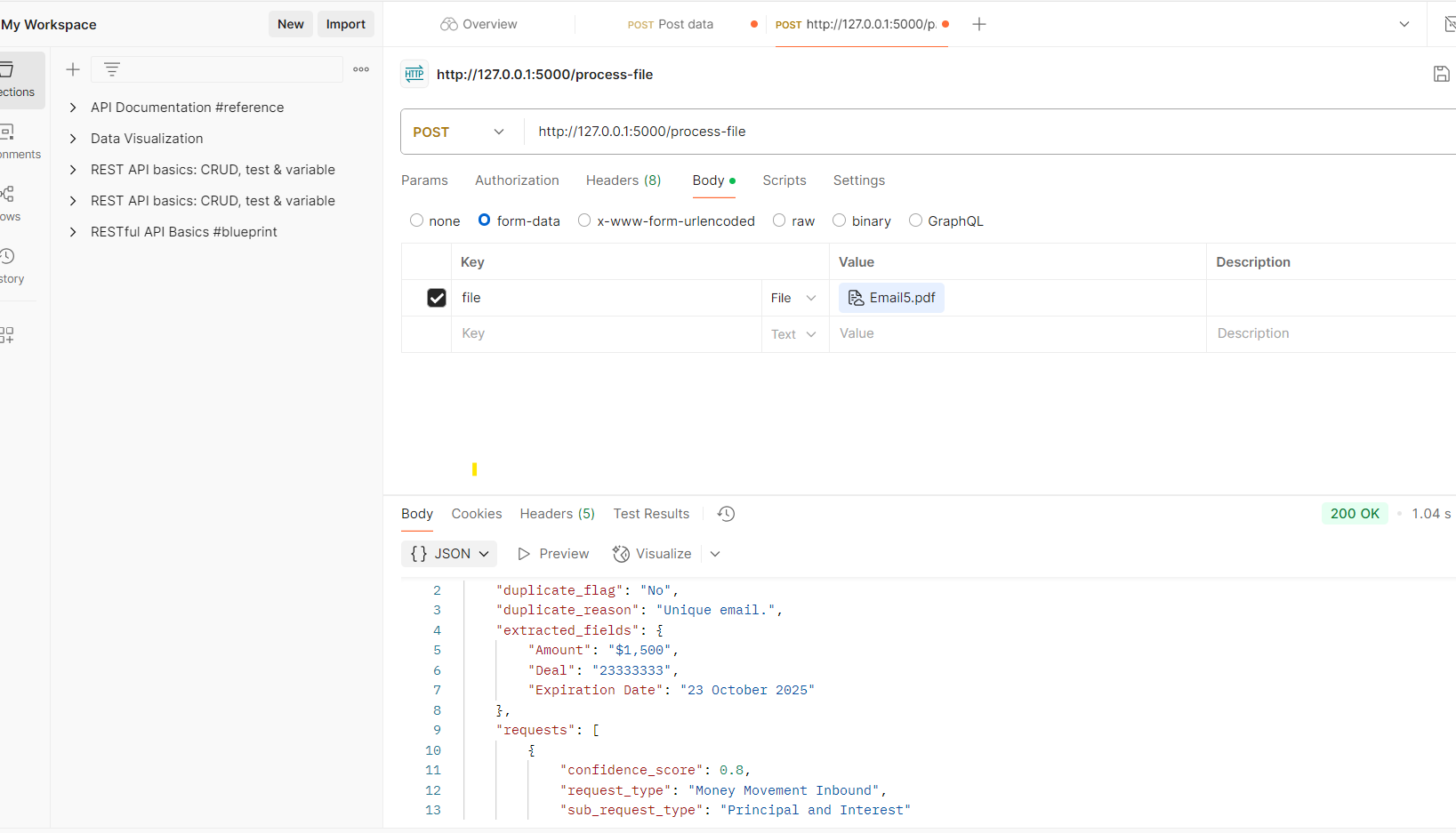
Step 4:

Get the final result:



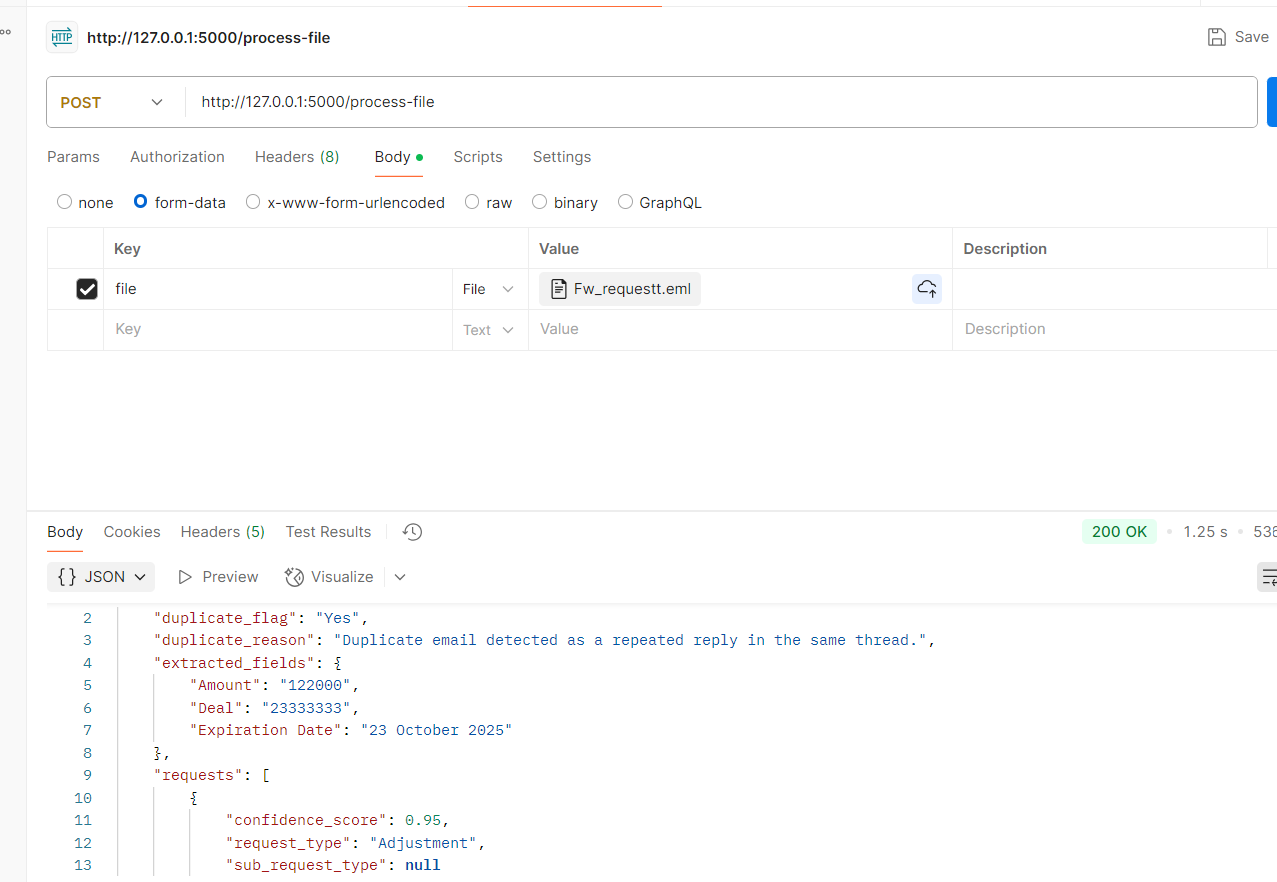
Step-5 (Either run code through cmd (step3)or Postman(step5)):

Run the code with Postman:



Step-6 Checking duplicate email as Sample for negative testing:





Demo Video and Document Git Repo Link:

<https://github.com/ewfx/gaied-ai-mark-tech/>