1. Write a program in python to scan a document or a jpg file(ex;- birth certificate, death certificate, Resume, invoices, PO..etc) using google or OpenAI API's and create a model to demonstrate the data extracted. Once they are successful in scanning one document they should do the same for bulk scanning of at least 100 documents and create a data model.

To accomplish this task, we will use the Google Cloud Vision API and OpenAI API. Here's a step-by-step solution:

Step 1: Set Up Environment

First, we need to install the necessary libraries:

bash

Edit

Copy code

pip install google-cloud-vision openai pandas numpy

Next, we need to set up our API keys for Google Cloud Vision and OpenAI.

Step 2: Initialize APIs

We'll initialize the clients for both APIs:

python

Edit

Copy code

import os

import pandas as pd

from google.cloud import vision

import openai

Set up API keys

os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'path/to/your/google-cloud-credentials.json'

openai.api_key = 'your-openai-api-key'

Initialize clients

client = vision.ImageAnnotatorClient()

Step 3: Single Document Scanning Function

We'll define a function to scan a single document using Google Cloud Vision API to extract text. Then, we'll use OpenAI API to refine and structure the extracted text into a data model.

python

Edit

Copy code

```
def scan_document(file_path):
    with open(file_path, 'rb') as image_file:
        content = image_file.read()
    image = vision.Image(content=content)
    response = client.text_detection(image=image)
    texts = response.text_annotations
    if texts:
        text = texts[0].description
        # Use OpenAI to refine and structure the text
        response = openai.Completion.create(
            engine="text-davinci-003",
            prompt=f"Extract structured data from the following text: {text}",
            max_tokens=150
        )
        structured_data = response.choices[0].text.strip()
        return structured_data
```

```
    return None
```

Step 4: Bulk Document Scanning

We'll create a function to iterate over a list of document paths and apply the single document scanning function. We'll store the results in a structured format, such as a DataFrame.

python

Edit

Copy code

```
def bulk_scan(file_paths):
    data = []
    for file_path in file_paths:
        structured_data = scan_document(file_path)
        if structured_data:
            data.append(structured_data)
    return data
```

Step 5: Data Model Creation

We'll define a schema for the data model based on the type of document (e.g., birth certificate, resume). We'll map the extracted data to the defined schema.

python

Edit

Copy code

```
def create_data_model(data):
    # Assuming the data is in a specific format, convert it to a DataFrame
    df = pd.DataFrame(data)
    return df
```

Step 6: Save and Visualize Data

We'll save the extracted and structured data to a file (e.g., CSV, JSON). Optionally, we can visualize the data using libraries like matplotlib or seaborn.

python

Edit

Copy code

```
def save_data(df, output_path):
    df.to_csv(output_path, index=False)
```

Example Usage

Let's scan a single document and then bulk scan a list of documents:

python

Edit

Copy code

```
file_paths = ['path/to/doc1.jpg', 'path/to/doc2.jpg', ..., 'path/to/doc100.jpg']

data = bulk_scan(file_paths)

df = create_data_model(data)

save_data(df, 'output.csv')
```

This program sets up the environment, initializes the APIs, defines functions for single and bulk document scanning, creates a data model, and saves the results. Adjust the file paths and API keys according to your needs.