**PROJECT** **TITLE: AI AGENT - SEARCH & RETRIEVE** **INFORMATION**

**AIM: You will create an AI agent that reads through a dataset (CSV or Google Sheets) and performs a web search to retrieve specific information for each entity in a chosen column. The AI will leverage an LLM to parse web results based on the user's query and format the extracted data in a structured output. The project includes building a simple dashboard where users can upload a file, define search queries, and view/download the results.**

* To create an AI agent as described, follow this step-by-step guide. The project will include a **Streamlit-based dashboard**, and leverage **SerpAPI** for web searches, **LangChain** for parsing results, and **OpenAI API** for NLP tasks.

**Project Summary**

The AI agent processes a dataset to:

1. Extract entities from a specific column.
2. Perform web searches for each entity using **SerpAPI**.
3. Parse and analyze web results using **LangChain** and **OpenAI API**.
4. Display and download structured output via a **Streamlit dashboard**.

**Setup Instructions**

1. **Clone the Repository(bash)**

git clone <https://github.com/Vineela-pilla/streamlit-app->

cd streamlit(app)

1. **Install Dependencies-** Create a virtual environment and install required Python packages(bash)

Python –m venv venv

Source venv/bin/activate

Pip install –r requirements.txt

1. **Set Up API Keys**

Get your **SerpAPI key** from [SerpAPI](https://serpapi.com/).

Get your **OpenAI API key** from [OpenAI](https://platform.openai.com/).

Create a .env file and add the keys:

SERP\_API\_KEY=b92bf5e8a2f5b5cf5fbbb6cd995ff93febdeef8ae999c4f2fc84a3ed9e6dc4b0

OPENAI\_API\_KEY=sk-proj-goYonWc17NiMAA\_Q- EhgT5oRf1JjltbosqXRSV6BnnhxgHYk8M3wAXRooibUeTUGdxDoQczoabT3BlbkFJSRzmCRdQogME1Sj\_mNqyFBx0hnHeqb453hAXbRkT41IFwwVH9ZPI4wN135wT6QvK1KeVrFNeUA

 GOOGLE\_SHEET\_API\_KEY=AIzaSyAXOYVPQ3mEG\_rOtv1CmPRS6pBwYk4Ki5g

4.**Run the app**

Streamlit run streamlit(app).py

**Usage Guide**

1. Upload a CSV or Google Sheets file.
2. Select a column for entity extraction.
3. Define a search query template (e.g., Find the latest news about {entity}).
4. View the results directly in the dashboard.
5. Download the structured output as a CSV.

**Key Features**

* **Streamlit Dashboard**: Intuitive UI for file upload and results visualization.
* **Dynamic Search Queries**: Customizable templates for each entity.
* **LangChain & OpenAI**: Robust parsing and analysis of search results.
* **Downloadable Output**: Export data in CSV format.

**Third-Party Tools**

* **Streamlit**: For building the user interface.
* **SerpAPI**: To perform Google searches programmatically.
* **LangChain**: To interact with the LLM for result parsing.
* **OpenAI API**: For natural language processing tasks.

**Code Snippet**

**1.Streamlit(app).py**

* **Data Input:**

Users can upload a CSV file or specify a Google Sheet name.The app loads and displays the data.

 **Dynamic Search**:

Users define a search query.The app integrates **SerpAPI** to retrieve results for each entity in the selected column.

 **Result Summarization**:

Search results are processed using **OpenAI API** for summarization. Summarized results are displayed and added to a structured output.

 **Downloadable Output**:

Users can download the final structured data as a CSV file.

**2. helpers.py**

* **Environment Variable Management**:
  + API keys are loaded securely using dotenv.
* **CSV Data Loading**:
  + The load\_csv\_data function reads a CSV file into a pandas DataFrame.
* **Google Sheets Integration**:
  + The load\_google\_sheet function connects to Google Sheets using gspread and retrieves data as a pandas DataFrame.
* **Web Search with SerpAPI**:
  + The search\_with\_serpapi function performs a search query and extracts organic results.
* **Interaction with OpenAI**:
  + The query\_openai function interacts with the OpenAI API to process prompts and return structured responses.

**3.Requirements.txt**

**steamlit**

**openai**

**pandas**

**langchain**

**serpapi**

**python-dotenv**

**gspread**

**oauth2client**

**4.(environmental variables).env**

SERP\_API\_KEY=b92bf5e8a2f5b5cf5fbbb6cd995ff93febdeef8ae999c4f2fc84a3ed9e6dc4b0

OPENAI\_API\_KEY=sk-proj-goYonWc17NiMAA\_Q-EhgT5oRf1JjltbosqXRSV6BnnhxgHYk8M3wAXRooibUeTUGdxDoQczoabT3BlbkFJSRzmCRdQogME1Sj\_mNqyFBx0hnHeqb453hAXbRkT41IFwwVH9ZPI4wN135wT6QvK1KeVrFNeUA

 GOOGLE\_SHEET\_API\_KEY=AIzaSyAXOYVPQ3mEG\_rOtv1CmPRS6pBwYk4Ki5g

**5.credentials.json**

It is used for accessing google sheets.