📘 Structured Extraction Pipeline

Objective

Build an **end-to-end LLM-powered structured extraction pipeline** that takes a short document, applies a prompt template, validates the output against a JSON schema, and saves the structured result.

💣 Architecture Overview

```
Document → Prompt Template → LLM Inference → Output Validation → Structured JSON → CSV Report
```

Stage	Description	Purpose
Input Validation	Checks for banned words & PII	Safety before inference
Prompt Template	Provides consistent structure & task definition	Guides the LLM
LLM Extraction	GPT-4o-mini model generates structured output	Information extraction
Schema Validation	Ensures JSON matches required fields	Structural integrity
Output Storage	Saves to CSV for analysis or downstream processing	Reusability

---

⚙️ Components

✂️ Input Validation

Rejects unsafe input containing: - Banned words (`password`, `confidential`, `credit card`) - PII patterns (emails, phone numbers)

✒️ Prompt Template

Guides the model to output JSON with keys:

```
{"title", "summary", "keywords", "category"}
```

🛏️ LLM Inference

- Uses **OpenAI GPT-4o-mini**
- Parameters:

- `temperature=0.0` → deterministic
- `top_p=0.9` → focused sampling
- `max_tokens=300` → concise output

🌂Schema Validation

Ensures the model output conforms to a strict JSON schema using `jsonschema`.

🌡️ Output Logging

- Valid results displayed as a DataFrame
- Stored in `structured_output.csv`

🧰Example Output

```
{
  "title": "Azure Cloud Services Overview",
  "summary":
"Azure helps businesses deploy and scale applications securely with AI and data
management capabilities.",
  "keywords": ["Azure", "Cloud", "AI", "Data", "Security"],
  "category": "Cloud Computing"
}
```

✅Schema Validated ✅Output Stored as CSV

🗜️How to Run

1. Open the notebook `structured_extraction_pipeline.ipynb` in **Google Colab**.
2. Add your API key securely:

```
from google.colab import userdata
userdata.set('OPENAI_API_KEY', 'sk-your-key-here')
```

3. Run all cells sequentially.
4. View structured output in CSV.

📊 Technologies Used

Category	Tools
LLM	OpenAI GPT-4o-mini
Validation	`jsonschema`, regex
Storage	Pandas CSV export
Environment	Google Colab

Best Practices

- Keep temperature = `0.0` for deterministic extraction
- Always apply input sanitization before calling LLMs
- Use JSON schema validation to ensure data reliability
- Integrate this pipeline with Airflow or LangChain for production

---

📁 Project Structure

```
/structured_extraction_pipeline/
├── structured_extraction_pipeline.ipynb
├── README.md
└── structured_output.csv
```

Author

Harsha — Principal Data Engineer & AI Systems Builder

- Focus: Scalable data and AI architectures (Azure, PySpark, GenAI)
- Project: LLM Pipeline Foundations (Days 8–14)