

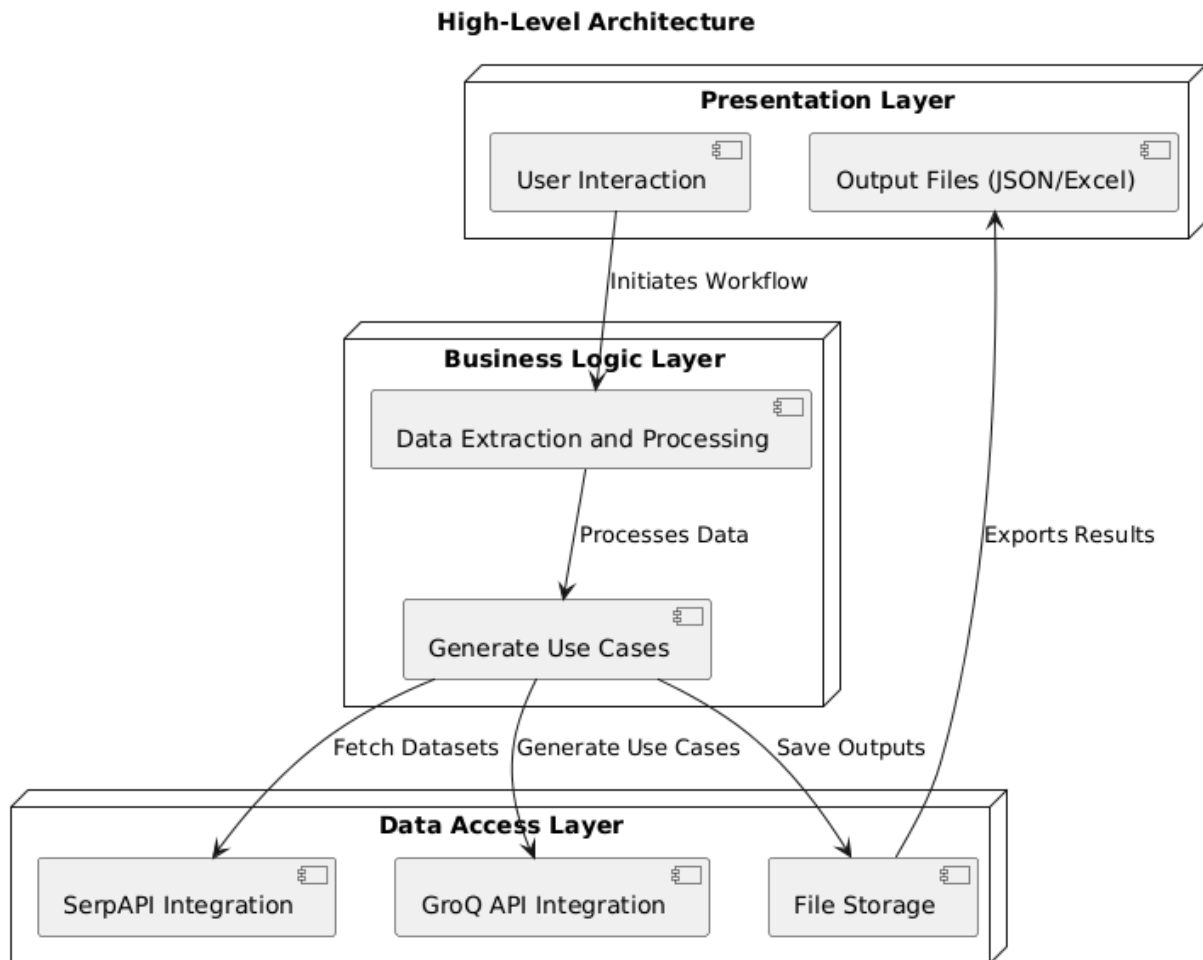
Documentation for the Application

1. Application Overview

This application automates **market research** and **use case generation** for a given company or industry. It integrates **API-driven agents** to:

1. **Conduct market research** about the company and its industry.
2. **Generate actionable AI/ML use cases** tailored to enhance operational efficiency, improve customer experience, and align with the company's goals.
3. **Fetch relevant datasets** for proposed use cases from Kaggle, GitHub, and HuggingFace.
4. Present results in structured formats (JSON and Excel), ensuring usability and clarity.

2. Application Architecture



The architecture follows a **multi-agent system**, inspired by the assignment requirements. Each component serves a specific role to achieve the end-to-end workflow.

Agents:

1. **Research Agent:** Fetches and analyzes industry and company-related data.
2. **Use Case Generator Agent:** Proposes AI/ML use cases using GroQ API.
3. **Dataset Fetcher Agent:** Searches for datasets related to the proposed use cases.

3. Workflow

1. Input:

- **Company Name:** The name of the company or industry to be analyzed.
- **Output Directory:** Path where results will be saved.

2. Process:

- The **Research Agent** queries the SerpAPI for relevant industry data.
- Extracted data is processed into **snippets**.
- The **Use Case Generator Agent** uses the snippets to generate AI/ML use cases.
- The **Dataset Fetcher Agent** searches for datasets relevant to each use case.
- Results are compiled into JSON and Excel formats.

3. Output:

- **JSON Files:** Raw research data, extracted snippets, and dataset links.
- **Excel File:** Use cases with corresponding dataset links in a structured table.

4. Key Functionalities

4.1 Research Industry

- **Function:** `research_industry(company_name)`
- **Description:** Queries for industry trends, business models, strategic focus areas, and AI adoption trends.
- **Output:** A dictionary with search results categorized by query.

4.2 Generate Use Cases

- **Function:** `generate_use_cases_with_groq(snippets)`
- **Description:** Uses GroQ API to create detailed use cases involving Generative AI, LLMs, and ML technologies.
- **Output:** A list of use cases with titles and detailed descriptions.

4.3 Fetch Dataset Links

- **Function:** `fetch_datasets_with_serpapi(keywords)`
- **Description:** Searches Kaggle, GitHub, and HuggingFace for datasets related to the given keywords.
- **Output:** A dictionary mapping dataset links to platforms.

4.4 Save Outputs

- **Functions:**
 - `save_to_file(data, file_name)`: Saves raw data to JSON.
 - `save_use_cases_to_excel_with_links(use_cases, datasets_file, file_name)`: Saves use cases with clickable dataset links in an Excel file.

5. Dependencies

- **Python Libraries:**
 - `requests`: For API calls.
 - `json`: To handle JSON data.
 - `openpyxl`: For Excel file generation.
- **APIs:**
 - **SerpAPI**: To fetch web search results.
 - **GroQ API**: For generating AI/ML use cases.

6. Error Handling

- **API Errors:**
 - Returns descriptive error messages in JSON for debugging failed API requests.
- **File I/O Errors:**
 - Handles file read/write errors to ensure robustness.

7. Application Execution

Main Function: `main(company_name, output_path)`

- Coordinates all agents to perform market research, generate use cases, and compile results.

Example Execution:

```
# Example Execution

if __name__ == "__main__":

    company_name = "infosys"

    output_path = r"C:\Users\user\Desktop\IP\reports"

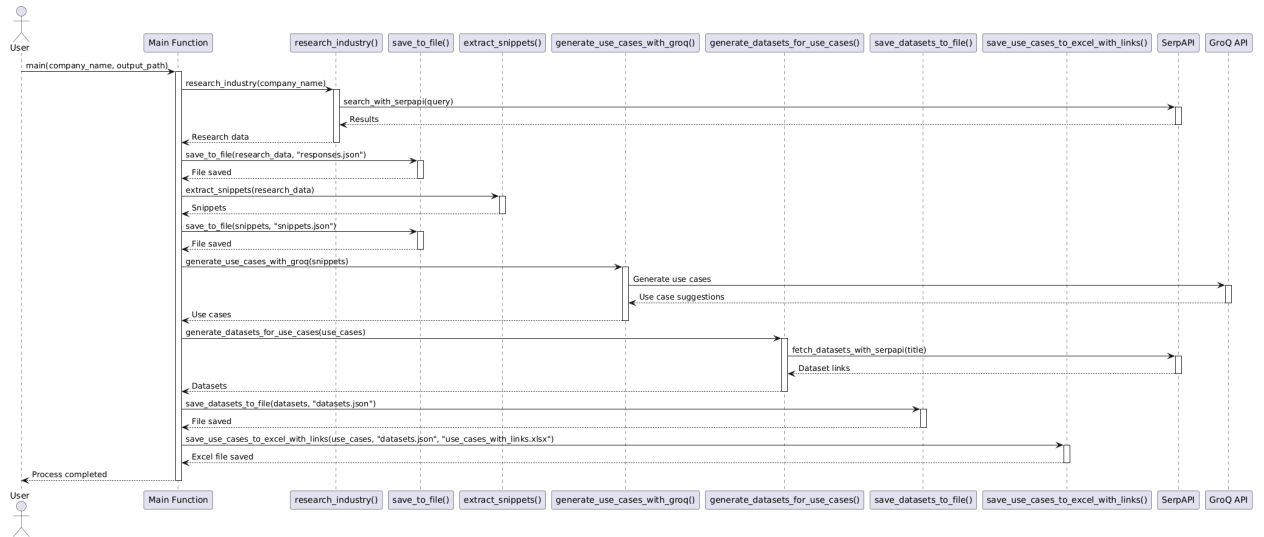
    main(company_name, output_path)
```

8. Deliverables

- **Source Code:** Fully annotated Python script.
- **Outputs:**
 - Research snippets (JSON).
 - Dataset links (JSON).
 - Use cases with dataset links (JSON / Excel).

- **Architecture Diagram:** A flowchart illustrating the multi-agent workflow (to be included in the final report).

[Image Link](#)



9. Future Enhancements

- **Integration:**
 - Support additional dataset platforms like Data.gov or UCI ML Repository.
- **Visualization:**
 - Add visual summaries of research insights and use cases using Streamlit or Gradio.
- **Error Resilience:**
 - Add retry logic for transient API/network errors.

10. References

- [Demo Video](#)
- [Source Code](#)