

**Department of Computer Science and Applications**

**Mini Project on Open-Source Technologies**

**RE-ZERO**

**FYMCA**

**Division – B.**

**FY 2024-2025**

**PROJECT MEMBERS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Name** | **PRN No.** | **Mobile No.** |
| 1 | Anshul Gupta | 1272240383 |  |
| 2 | Philip B Kuriackal | 1272240297 |  |

**PROBLEM STATEMENT:**

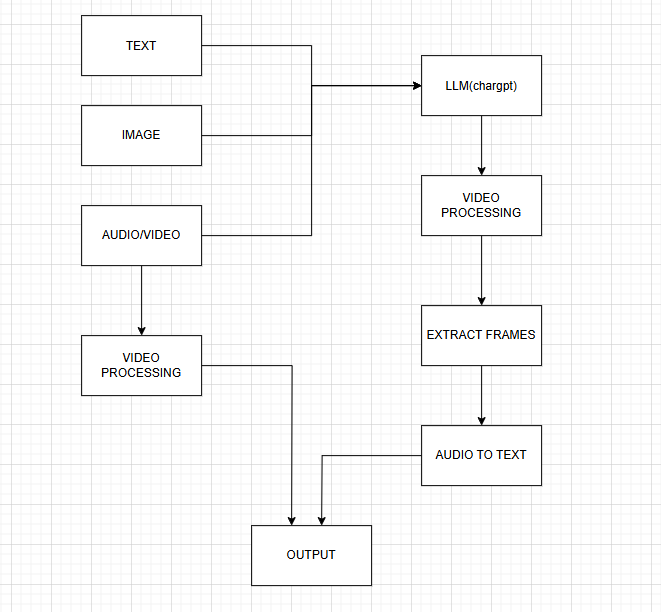
Current artificial intelligence solutions often lack the autonomy, adaptive decision-making, and long-term memory required to independently tackle complex, multi-step tasks in dynamic environments. This limitation leads to significant human oversight, inefficient workflows, and a reduced ability to scale AI applications beyond predefined, reactive operations. Organizations and individuals seeking true automation for intricate challenges are therefore constrained by the necessity of constant human intervention and the inherent inability of conventional AI systems to reason, plan, and learn effectively over extended interactions or unforeseen circumstances. This project addresses the critical need for an Agentic AI system that can perceive, reason, act, and learn autonomously, thereby unlocking advanced automation capabilities, minimizing human intervention, and enabling AI to solve more sophisticated real-world problems

**PROJECT SCOPE:**

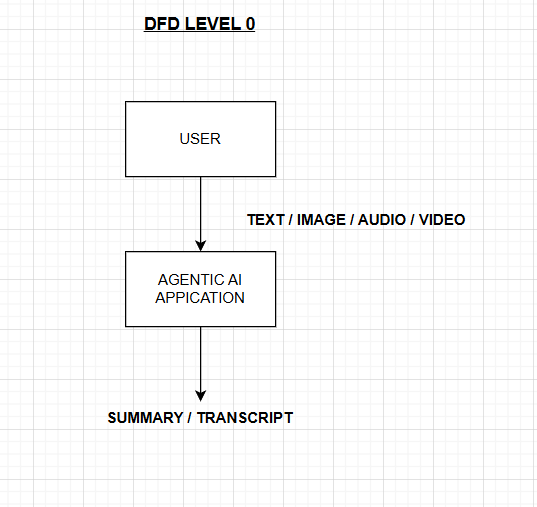
The project will focus on developing a foundational understanding and practical implementation of Agentic AI. This includes:

* **Core Agentic Principles:** Implementing components like perception, reasoning (via LLMs), action (tool use), and learning.
* **Tool Integration:** Seamlessly connecting AI agents with various external tools and APIs for diverse functionalities.
* **Orchestration:** Managing and coordinating multiple AI agents to achieve larger, more complex objectives.
* **User Interface:** Providing an intuitive interface for interacting with and observing agent behaviour.
* **Proof of Concept:** Developing a demonstrative application showcasing the capabilities of the Agentic AI system.

**FLOW OF THE SYSTEM**

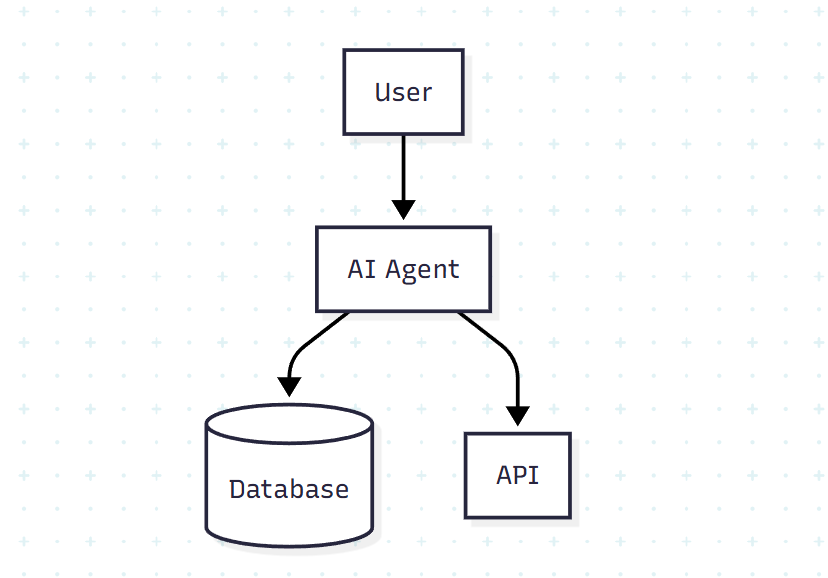
****

**DIAGRAMS (UML, ER, DFD, ACTIVITY DIAGRAM)**

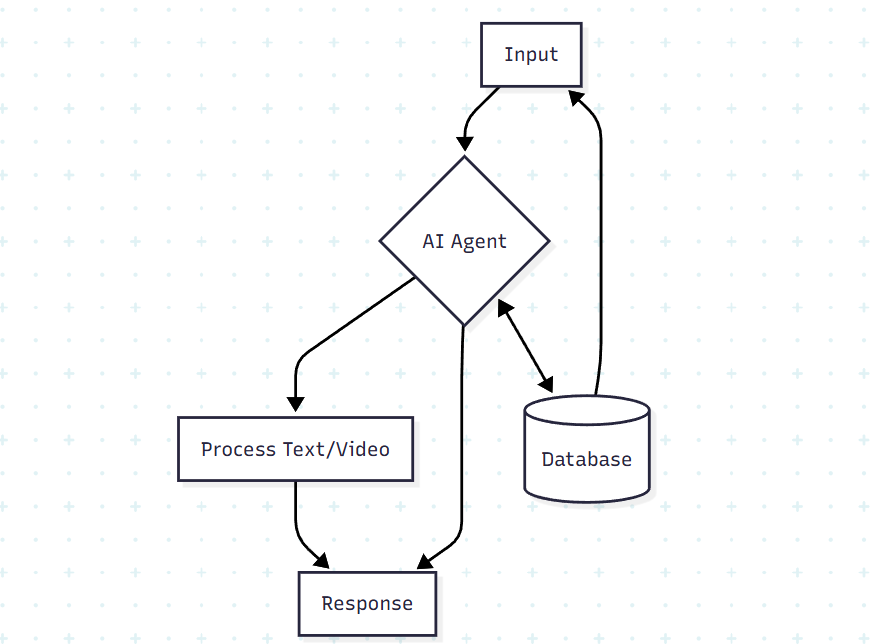
****

**ER DIAGRAM:**

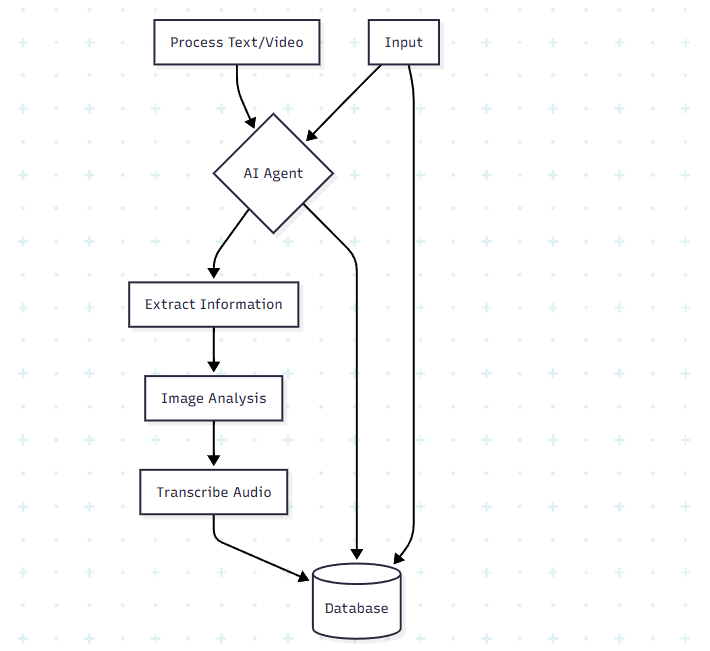
**Level 0**

****

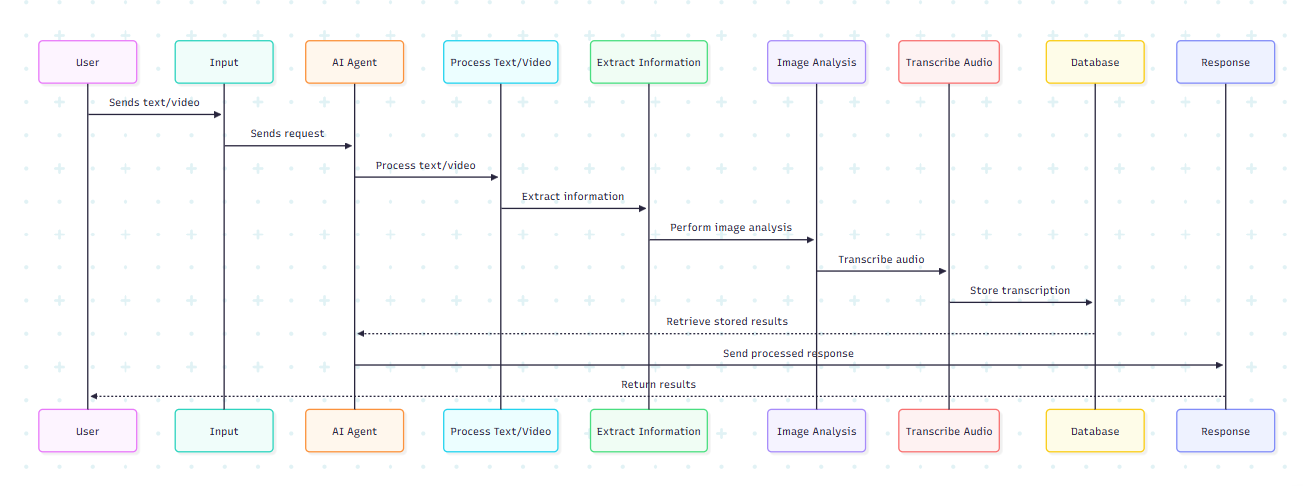
**Level 1**

****

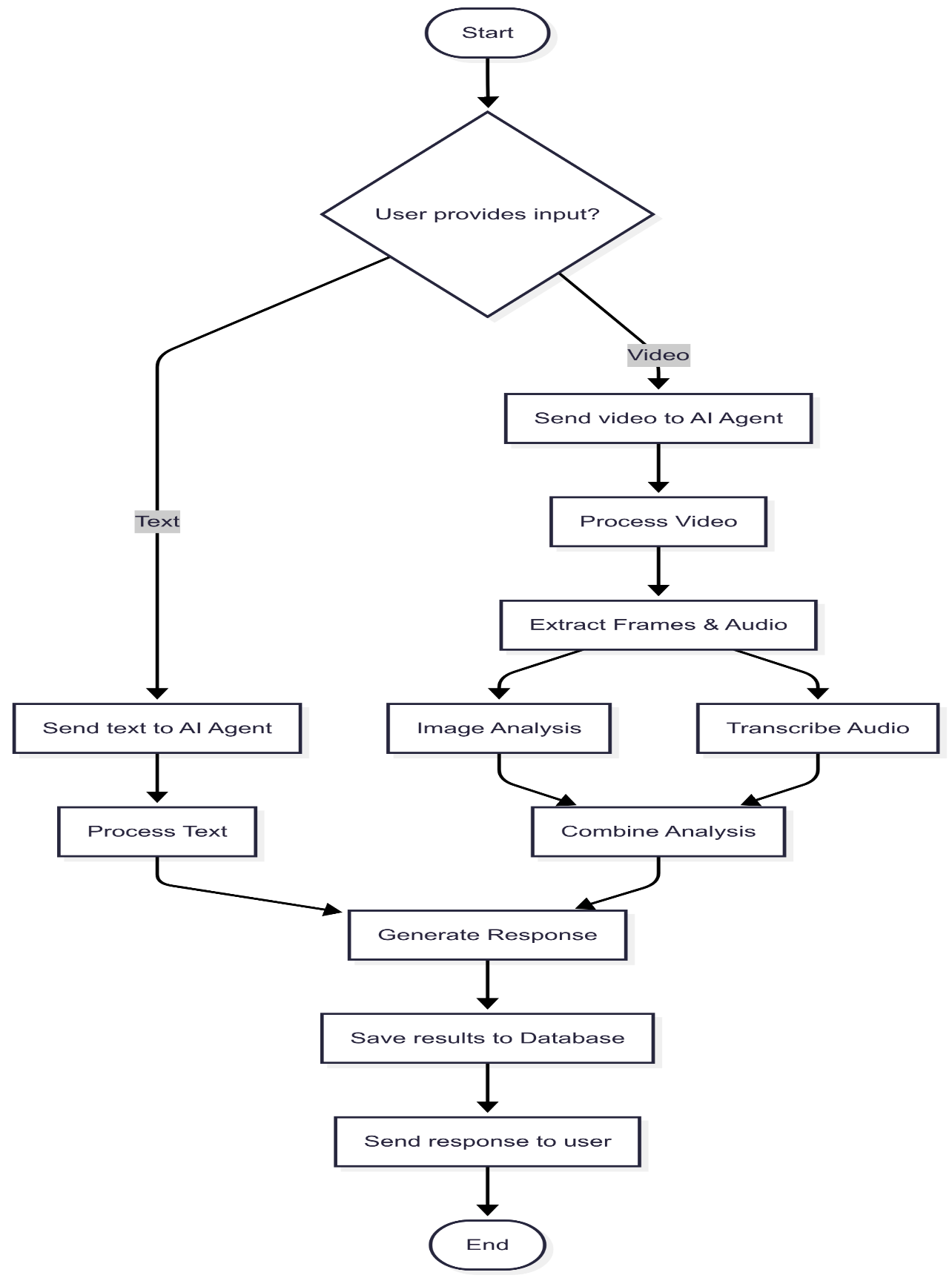
**Level 2**

****

**SEQUENCE DIAGRAM:**

****

**ACTIVITY DIAGRAM:**

**MODULE DESCRIPTON:**

### 1. User Interface Module

* **Purpose:** To provide a simple and intuitive point of interaction for the user. It serves as the frontend for the entire system, enabling users to submit requests and view the agent's progress and final outputs.
* **Core Component:** **Chainlit**.
* **Functionality:** Handles all aspects of the user interface, from rendering chat messages and multimedia outputs (images, videos) to capturing user input and managing conversational flow. It abstracts away the complexities of web development, allowing for rapid deployment of the agent's interface.

### 2. Agentic Core & Orchestration Module

* **Purpose:** This is the central processing and control unit of the system. It is responsible for managing the lifecycle of AI agents, delegating tasks, and facilitating collaboration among them to achieve complex, multi-step goals.
* **Core Components:** **CrewAI** and **LangChain**.
* **Functionality:** **CrewAI** orchestrates the multi-agent workflow by defining roles, tasks, and the collaboration process. It delegates a user's request to a team of agents who work together. **LangChain** provides the foundational framework for building the individual agents, handling their memory, tool access, and the connection to Large Language Models (LLMs).

### 3. LLM Providers Module

* **Purpose:** To provide the reasoning and natural language processing capabilities for the AI agents. This module gives the agents the "brain" to understand human requests, formulate plans, and make decisions.
* **Core Components:** **OpenAI API (ChatGPT)** and **Hugging Face**.
* **Functionality:** Agents interact with these APIs to generate text, understand context, perform advanced reasoning, and access a wide variety of specialized AI models. The choice between providers depends on the specific task, cost, and model requirements.

### 4. Tool Integration Module

* **Purpose:** To extend the capabilities of the agents beyond pure language processing by enabling them to interact with the external world. This module consists of various tools that agents can "call" to perform specific actions.
* **Core Components:** **Pandas**, **MoviePy**, **Pillow**, and **Whisper**.
* **Functionality:**
  + **Pandas:** The data manipulation tool, used for data analysis, cleaning, and structuring.
  + **MoviePy/Pillow:** Multimedia tools for creating, editing, and processing videos and images.
  + **Whisper:** The audio transcription tool, enabling agents to process and understand spoken language.

### 5. Data & Storage Module

* **Purpose:** To provide the necessary memory and data persistence for the agents to function effectively. This includes storing short-term conversational context, long-term knowledge, and large files.
* **Core Components:** **Vector Databases** (e.g., ChromaDB), **Relational Databases** (e.g., PostgreSQL), and **Cloud Object Storage** (e.g., AWS S3).
* **Functionality:**
  + **Vector Databases:** Serve as the agent's long-term memory, storing knowledge as semantic embeddings for efficient retrieval-augmented generation (RAG).
  + **Relational Databases:** Used for structured data like user profiles, project logs, and agent configuration.
  + **Cloud Object Storage:** Stores large, unstructured files such as generated images, videos, and raw data sets.

**Hardware Requirements:**

* Processor : 64-bit, Quad-core i5/i7 or Ryzen 5/7
* RAM : At least **8 GB** (16 GB better)
* Hard disk : 100 GB free space (prefer SSD)
* Monitor : Full HD or better for comfort
* GPU : NVIDIA GPU

**Software requirements:**

* Operating system : Windows 10/11
* Language : Python 3.10+
* Front end : CHAINLIT
* Back end : FastAPI / OpenAI (API)
* Database : PostgreSQL

**LIMITATION:**

 **Computational Resources:** Running advanced LLMs and multi-agent systems can be computationally intensive, requiring sufficient hardware or cloud resources.

 **Cost of APIs:** Reliance on commercial APIs like ChatGPT might incur costs depending on usage.

 **Complexity of Debugging:** Debugging autonomous and multi-agent systems can be challenging due to their non-deterministic nature and complex interactions.

 **Ethical Challenges:** Ensuring fairness, transparency, and preventing unintended biases or emergent behaviors remains a significant challenge.

 **Security Concerns:** Integrating with external APIs and allowing agents to take actions introduces security vulnerabilities if not carefully managed.

 **Data Dependency:** The quality and availability of data for agent training and operation directly impact performance.

**CONCLUSION:**

This Agentic AI project aims to build a powerful and flexible system for developing autonomous intelligent agents. By combining cutting-edge AI/ML models, robust frameworks, and intuitive interfaces, the project seeks to unlock new possibilities for automation, complex problem-solving, and intelligent decision-making. While challenges related to computational resources, ethical considerations, and debugging exist, the potential for future enhancements and real-world applications makes this an exciting and impactful endeavor.

Sign of each team members.