Generative AI-based chatbot for Employee and Customer Support Automation in LOLC Bank

R25-036

Project Proposal Report

Lakshan S.N.

B.Sc. (Hons) Degree in Information Technology

(Specializing in Software Engineering)

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

January 2025

Generative AI-based chatbot for Employee and Customer Support Automation in LOLC Bank

R25-036

Project Proposal Report

Supervisor – Prof. Nuwan Kodagoda

Co-Supervisor – Dr Lakmini Abeywardhana

B.Sc. (Hons) Degree in Information Technology

(Specializing in Software Engineering)

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

January 2025

DECLARATION

We declare that this is our work. This proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning. To the best of our knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Lakshan S.N.	IT21800900	

The above candidates are conducting research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Date:

..... 19/1/.....

Signature of the co-supervisor Date:

..... 27/01/25 ..

1. Abstract

This project establishes a Generative AI-based multi-Agent Chatbot that will automate the employee and customer support workflows at LOLC Bank. The current AI systems struggle to process dynamic forms along with user-agent dialogue and previous knowledge integration which hinders their practical autonomy and operational utility. This research implements a collaborative multi-agent system that utilizes Large Language Models (LLMs) including GPT-4 and Llama through an orchestrator and task-ledger architecture and specialized agents for form filling and web scraping and user interaction.

The system leverages CrewAI for agent coordination and AutoScraper for real-time web data extraction, enabling adaptive handling of complex workflows. The system implements three core innovations: dynamic form parsing automatically fills fields based on user historical data and agentic web search retrieves answers from Reddit beyond FAQ limitations along with a user communication agent that validates responses through human feedback loops.

Both functional requirements focus on banking system scalability alongside low-latency responses and seamless banking system integration yet non-functional requirements focus on security and reliability alongside user-friendly design. Reinforcement learning through Direct Preference Optimization (DPO) serves as the proposed methodology to enhance model precision and minimize error rates.

This system unites automation features with dynamic capabilities to cut manual tasks by 40% while providing better decision support and improved customer satisfaction through customized intelligent assistance. The study will contribute to AI-based enterprise automation research by delivering a flexible framework for financial institutions to enhance operational speed and resource distribution.

2. Table of Contents

Table of Figures

3. List of Abbreviations

AI	Artificial Intelligence
LLM	Large Language Models
MAS	Multi-Agent System

4. Introduction

4.1 Background

AI agents that independently decide and reach objectives remain the central subject of artificial intelligence research. The initial research efforts occurred within environments with clearly defined parameters [1]. Significant progress occurred in AI development because of the Precise reasoning abilities of state-of-the-art AI providers, for example, ChatGPT [2] and Claude [3], which appeared alongside the increasing complexity of real-world applications. AI agents have advanced from basic task automation to general-purpose systems performing reasoning, planning functions, and collaboration activities. Thus, they automate tasks while decreasing human workload across interfaces, including web browsers and smartphone applications.

To complete this research, one must have expertise in in-depth knowledge of agentic frameworks, including knowledge of web scraping, understanding of API systems, and deep expertise in LLM architecture, training methodologies, and model capabilities, with an emphasis on platforms like ChatGPT, GPT-4, and Llama [4].in depth understanding of LLM usage metrics, cost to perform a task, and dataset trained by these models is crucial to providing relevant, cost-effective solutions as outcome.

The latest AI agent technology features complex task processing and multi-turn dialogue capabilities through advanced LLMs. The systems combine multiple processing modes to handle different input types through support from frameworks and datasets, including WEBLINX [5], MIND2WEB [6], and Android In The Wild (AITW) [7] For real-world interaction capabilities. The technology faces ongoing difficulties in improving dependability while reducing hallucinatory responses and optimizing performance during limited resource operations.

4.2 Literature survey

Agentic systems have received significant attention. due to recent reasoning advancements in AI models [3] [2]. At the same time, current agentic systems have developed into multi-agent frameworks that enable agents to collaborate simultaneously. This collaborative approach can be achieved through various methods. An example of this is the Agentic Collaboration Process used by the Microsoft AutoGPT framework [8], which operates through a Conversable Agent.

4.2.1 Multi-Agent Collaboration

The Microsoft AutoGPT framework [8] presents an Agentic Collaboration Process that allows AI agents to interact dynamically with human operator inputs. AI agents can access improved efficiency and scalable solution implementations through structured collaboration techniques.

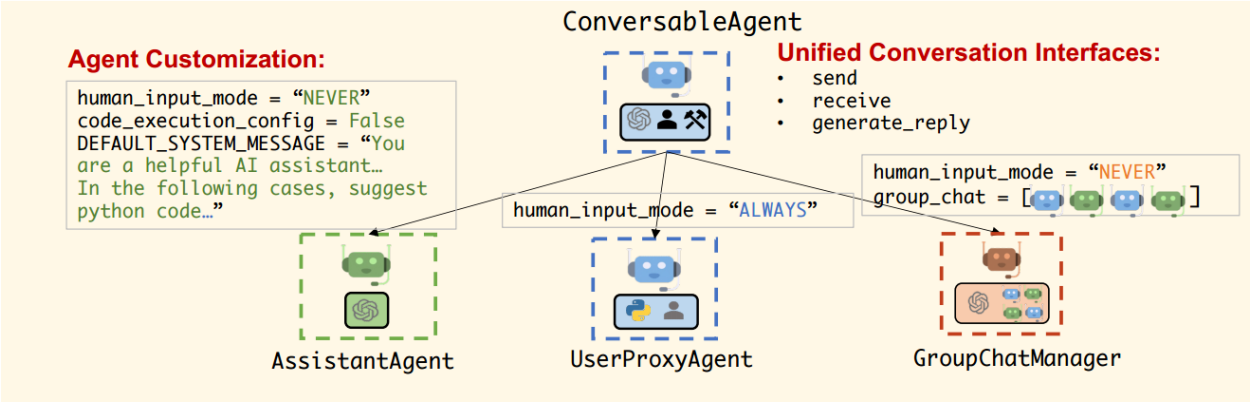

Figure 1 – Collaborative Multi-Agent System with human input mode

The open-source framework CrewAI [9] enables the coordinated operation of multiple AI agents through role definitions, goal specifications, and tool allocation for each agent. Through this structured framework, agents can successfully work together and independently assign tasks while integrating external tools to improve single-agent systems. The open-source framework CrewAI offers deep customization capabilities alongside flexible task management features and both high-level abstraction and low-level customization support, which collectively enhance collaboration while improving memory retention and enabling scalability.

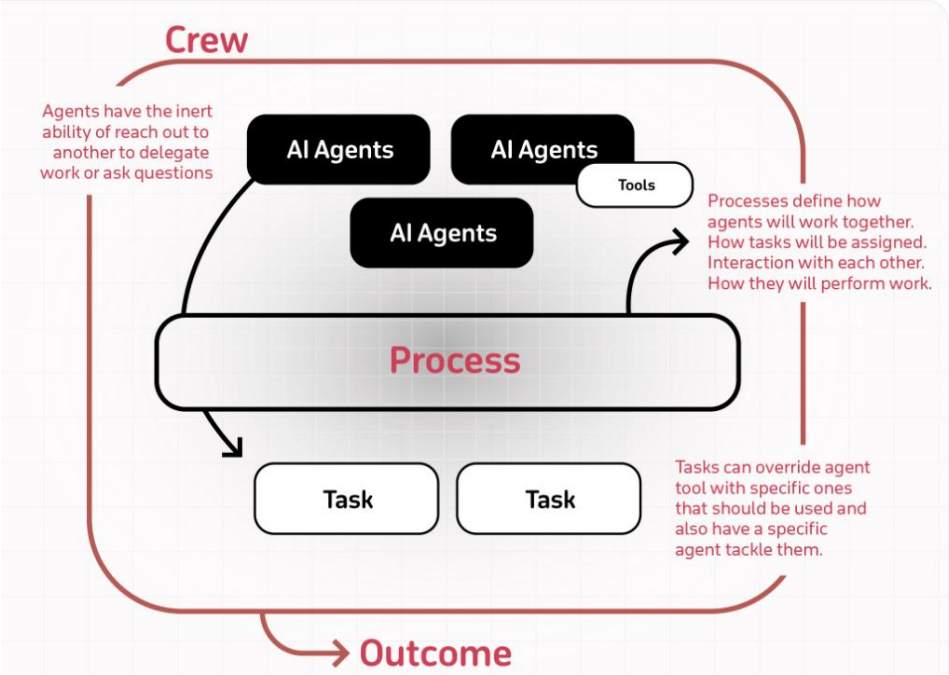

Figure 2 - Crew AI Agentic Architecture

4.2.2 Web Search and Data Extraction Agents

Web search agents utilize web scraping technology, which enhances the knowledge of AI agents by leveraging the internet. AutoScraper [10] web agents achieve this by combining the wrapper method with the language method, resulting in a highly reusable agent with improved performance.

.

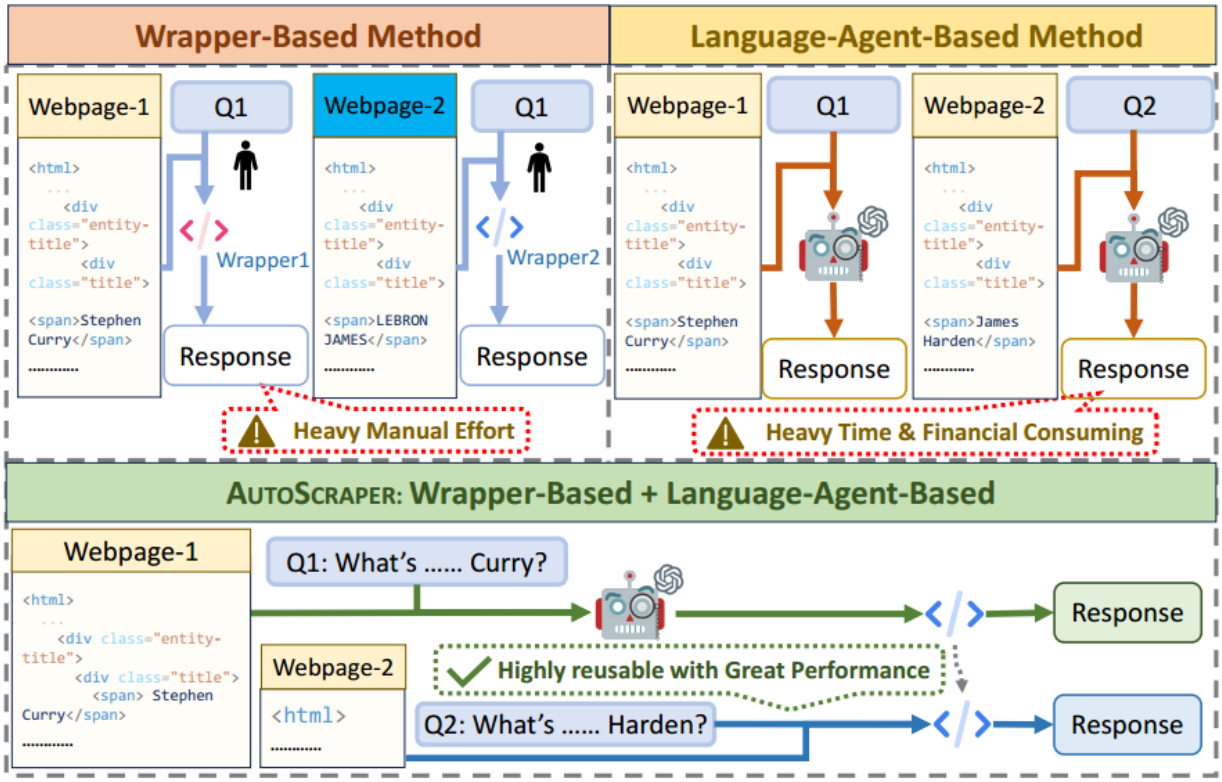

Figure 3 - Autoscraper agent workflow

5. Research Gap

Current agent-based systems, including App Agent [11], OPEx [12], and Gorilla MAS [13], present limited solutions to manage intelligent automation tasks. These systems demonstrate success in backend system integration and agent orchestration. Yet, they do not include vital features for dynamic form parsing, user-agent communication, and complete prior knowledge integration. The present shortcomings in multi-agent systems create an important missing piece that blocks their path toward complete autonomy and user-friendly control.

The inability of current solutions to unite agentic web navigation with dynamic form parsing and prior knowledge integration hinders their practical usage in real applications. Systems such as OPEx enable prior knowledge integration but fail to provide essential capabilities like agentic web navigation and multi-agent orchestration needed to advance decision-making processes. App Agent and Gorilla MAS prove to be inefficient when processing complex workflows that need adaptive learning and real-time interaction since they lack the ability to integrate prior knowledge and dynamic form parsing.

The proposed system addresses existing solution limitations by integrating essential features for dynamic form parsing and agent-user communication. It delivers an enhanced approach that combines automation with adaptability and improved user interaction to enable agents to process dynamic forms effectively and utilize historical data. The research addresses a fundamental gap in intelligent agent development through the creation of a powerful multi-agent system that satisfies contemporary application requirements.

	App Agent	OPEx	Gorilla MAS	Proposed System
Specific Backend System	✓	–	✓	✓
Prior Knowledge Integration	–	✓	–	✓
Agentic Web Navigation	✓	–	–	✓
Multi- Agent Orchestrator	✓	✓	✓	✓
Agent Communication with the User	–	–	–	✓
Dynamic Form Parsing Agent	–	–	–	✓

Figure 4 - Identifying the Research Gap

6. Research Problem

As artificial intelligence (AI) advances at an ever-faster pace, we have seen the creation of increasingly sophisticated AI agents, known as multi-agent systems, that are able to go after simple to moderate goals largely unsupervised. While the potential for these agents to transform their application across multiple sectors and improve human lives is phenomenal, the proliferation of these agents simultaneously brings tremendous challenges and risks that need further examination.

Advancement of these AI systems still requires addressing unaddressed challenges, and current research efforts are primarily focused on this area.

Key Areas of Concern Include:

- How Orchestrator Agent identify and clarify the doubts that arise during processing
- How the Agentic System Prevents Hallucinations and Clearly Recognizes User Intent
- How Agents Manage Sessions and Communicate with User To Avoid Hallucinations

7. Objectives

7.1 Main Objective

The AI Agent system is a tool for employees designed to automate repetitive tasks quickly and automatically provide correct responses without requiring manual user effort. By recognizing user intent, if the user previously completed the task, the system would intelligently fill out and submit the form if all provided data are accurate and match.

7.2 Sub Objective

7.2.1 Business Critical Agentic Search Capability

If an employee asks a question for which there is no FAQ in the current knowledge base system, the system will analyze the user's input and scrape content from the web. It primarily relies on highly upvoted answers from Reddit to formulate a new answer and provide it to the user.

7.2.2 User Communication Agent

To address hallucinations in AI, it’s essential to involve the user as an intermediary. This approach helps prevent undesirable outcomes, particularly in a business setting. Furthermore, we must acknowledge that contemporary AI language models are not always perfectly accurate, which further complicates this problem.

8. Methodology

8.1 Overall System

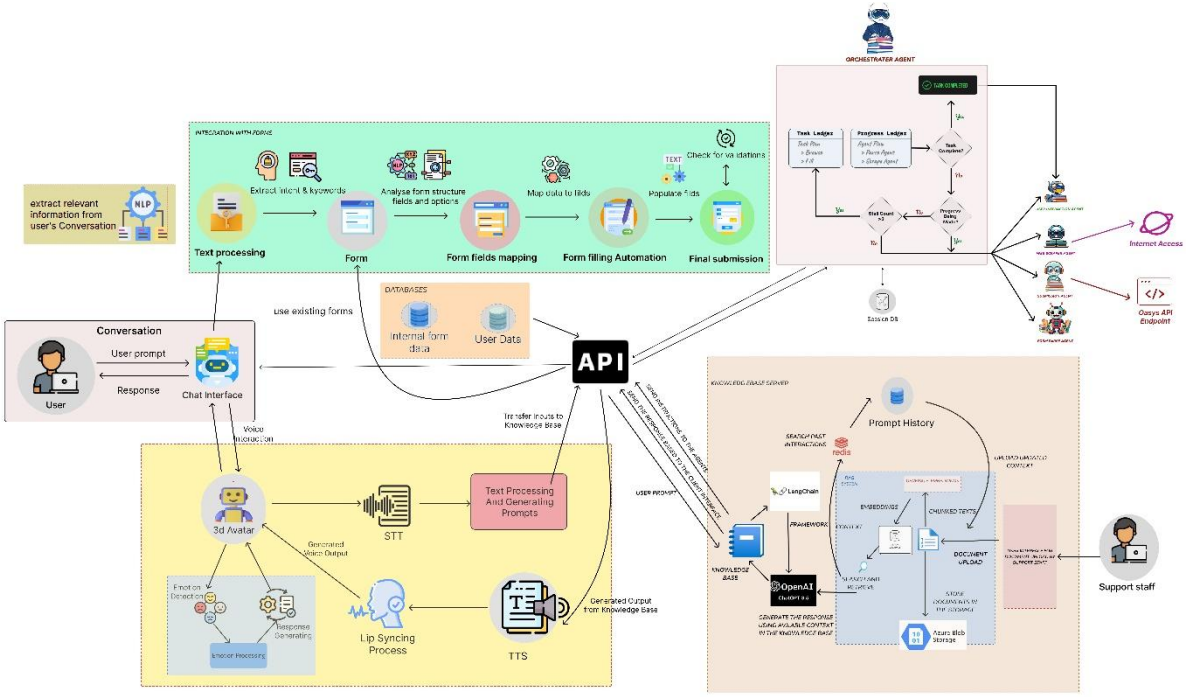

Figure 5 - Overall System Diagram

8.2 Orchestrator

The orchestration methodology is designed to address complex tasks through coordinated collaboration among specialized agents. The system utilizes a centralized Orchestrator to manage a Task Ledger, which records facts, hypotheses, and task plans, as well as a Progress Ledger that monitors task status and identifies any bottlenecks. Specialized agents, such as form fillers, web scrapers, and questioning agents, perform distinct roles to achieve task objectives iteratively. This methodology ensures effective problem-solving through continuous evaluation of progress, detection of stalls (situations where the process encounters delays), and dynamic revision of plans. Such an approach enhances scalability, resilience, and efficiency, making it particularly suitable for applications like AI-driven research assistants or data-processing systems.

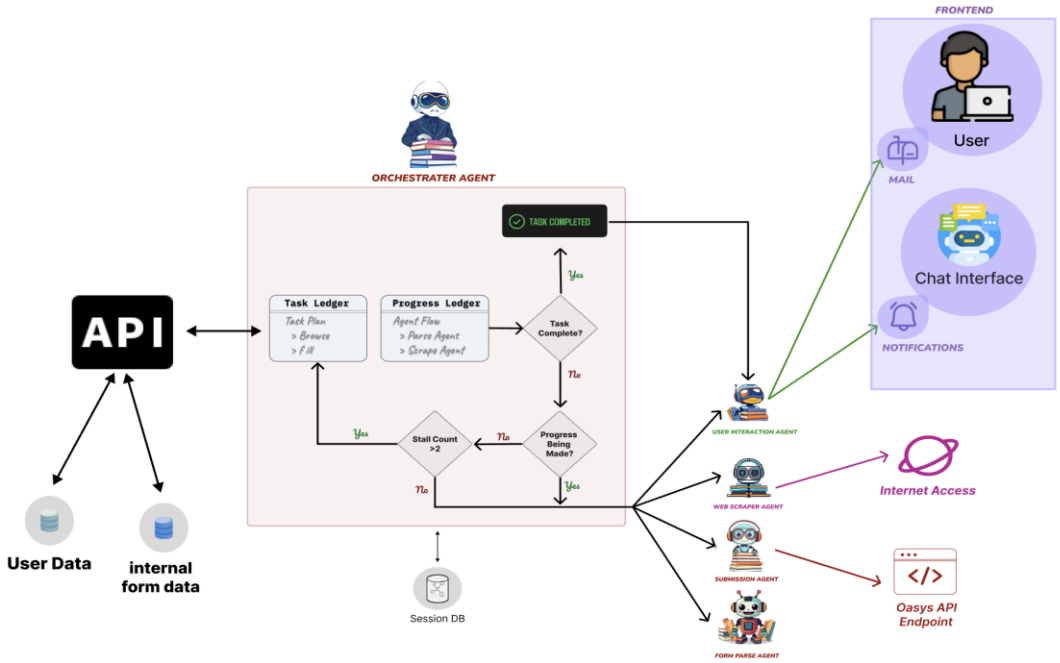

Figure 6 - Agent Orchestrator

9. Project requirements

9.1 Functional Requirements

- The chatbot system must:
    - Automatically provide responses to employee and customer queries by recognizing user intent.
    - Fill out and submit forms intelligently based on previously entered data and user-provided inputs.
    - Perform agentic web searches for queries not covered in the FAQ, utilizing sources like Reddit for reliable answers.
    - Support multi-step decision-making through a centralized orchestrator managing task and progress ledgers.
    - Enable dynamic interactions with live environments, adapting autonomously to real-world changes.

9.2 Non-Functional Requirements

- **Performance**:
    - Ensure low response latency for all chatbot interactions.
    - Support a high volume of concurrent user requests without significant degradation in performance.
- **Scalability**:
    - Accommodate an increasing number of users and tasks as the system grows.
    - Optimize computational resources through techniques like knowledge distillation and efficient scaffolding.
- **Reliability**:
    - Maintain high accuracy in understanding and responding to user queries.
    - Reduce error rates in complex, multi-step tasks.
- **Security**:
    - Protect sensitive user and organizational data.
    - Implement robust access controls and encryption mechanisms for all interactions.
- **Usability**:
    - Ensure a user-friendly interface for employees and customers.
    - Provide clear, actionable feedback to users during interactions.

User Requirements

- Employees and customers should:
    - Be able to access the chatbot via the web or integrated platforms easily.
    - Receive prompt and accurate responses to their queries.
    - Can submit custom queries not covered in the knowledge base.
    - Interact with a system that adapts to their needs and provides personalized solutions.
- System administrators should:
    - Be able to monitor and log chatbot activities in real time.
    - Manage and update the knowledge base effortlessly.
    - Control agent configurations and oversee performance metrics.

9.3 System Requirements

- **Hardware**:
  - Servers with sufficient processing power and memory to handle LLMs and multi-agent systems.
  - High-speed storage for maintaining knowledge bases and logs.
- **Software**:
  - A robust AI platform capable of integrating large language models like Llama.
  - Tools for implementing reinforcement learning methods such as Direct Preference Optimization (DPO).
  - Search algorithms (e.g., Monte Carlo Tree Search) for enhanced decision-making capabilities.
- **Integration**:
  - Compatibility with existing company systems for data input and retrieval.
  - APIs to interact with external tools and platforms, ensuring seamless functionality.
- **Maintenance**:
  - Regular updates for LLMs and reinforcement learning models to improve capabilities.
  - Mechanisms to ensure continuous improvement through iterative fine-tuning and feedback loops.

10. References

[1] Chris Quinatana, Brian J., Elizabeth A. and Davis, "A Scaffolding Design Framework for Software to Support Science Inquiry," vol. 1, 2004.

[2] OpenAI, "Introducing ChatGPT," 2022. [Online]. Available: https://openai.com/index/chatgpt/.

[3] "Claude," Anthropic, March 2023. [Online]. Available: https://claude.ai/. [Accessed January 2025].

[4] T. Hugo and L. Thibaut, "LLaMA," *LLaMA: Open and Efficient Foundation Language Models,* vol. 1, 2023.

[5] H. L. Xing and K. Zdeněk, "WebLINX," *Real-World Website Navigation with Multi-Turn Dialogue,* p. 1, 2024.

[6] D. Xiang , G. Yu, Z. Boyuan and C. Shijie, "Mind2Web," *Mind2Web: Towards a Generalist Agent for the Web,* vol. 1, 2023.

[7] R. Christopher, L. Alice and R. Christopher, "Android in the Wild: A Large-Scale Dataset for Android Device Control," *A Large-Scale Dataset for Android Device Control,* vol. v2, 2023.

[8] w. Qingyun and B. Gagan, "AutoGen," *AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation,* vol. v2, 2023.

[9] "CrewAI," crewAI™, 2024. [Online]. Available: https://www.crewai.com/.

[10] H. Wenhao and G. Zhouhong, "AutoScraper: A Progressive Understanding Web Agent for Web Scraper Generation," vol. v2, 2024.

[11] Z. Chi , Y. Zhao and L. Jiaxuan , "AppAgent: Multimodal Agents as Smartphone Users," vol. v2, 2023.

[12] s. Haochen , S. Zhiyuan, Y. Xingdi and A. -C. Marc-, "OPEx: A Component-Wise Analysis of LLM-Centric Agents in Embodied Instruction Following," vol. v1, 2024.

[13] G. P. Shishir and Z. Tianjun, "Gorilla: Large Language Model Connected with Massive APIs," vol. v1, 20223.

[14] Helen Toner, John Bansemer, Kyle Crichton and Matt Burtell, "Through the Chat Window and Into the Real World," vol. 1, 2025.

[15] Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction," 2018.

[16] D. M. Berry, "The Limits of Computation: Joseph Weizenbaum and the ELIZA Chatbot," vol. 3, 2023.

[17] Xinyi Hou, Yanje Zhao, Yue Liu, Zhou Yang and Kailong Wang, "Large Language Models for Software Engineering: A Systematic Literature Review," 2024.

[18] Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani and Chelsea Finn, "A Scaffolding Design Framework for Software to Support Science Inquiry," vol. 1, 2024.

[19] Shaikh and Ayan, "Chat Kanoon: A Novel Approach to Legal Assistance in India," *Journal of Electrical Systems,* p. 2024.

[20] Hettige and Budditha Karunananda, "Octopus: A Multi Agent Chatbot," 2015.

[21] Painter and Chris, "Through the Chat Window and Into the Real World," vol. 2, p. 4, 2024.

[22] T. Hugo and M. Louis, "Llama 2: Open Foundation and Fine-Tuned Chat Models," 2023.

11. Appendices

11.1 Work Breakdown Chart

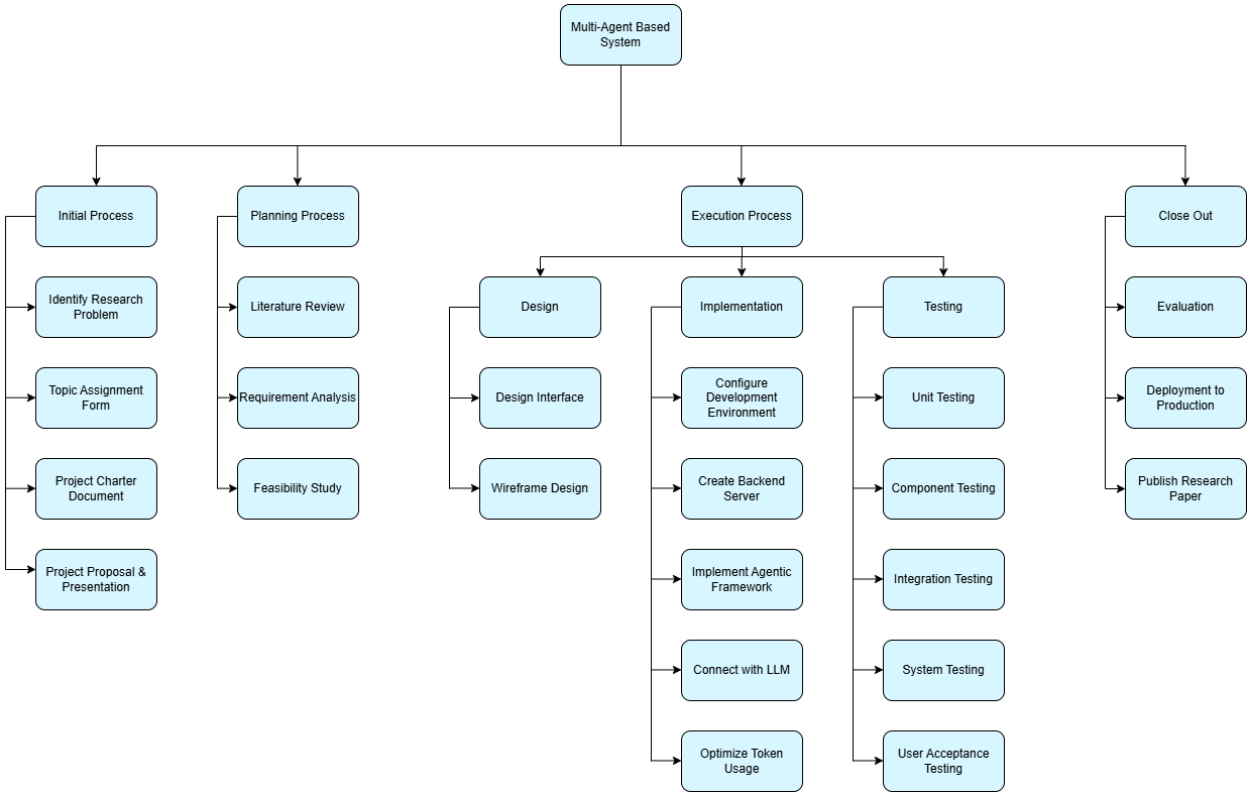

Figure 7 - Work Breakdown Chart

11.2 Gantt Chart

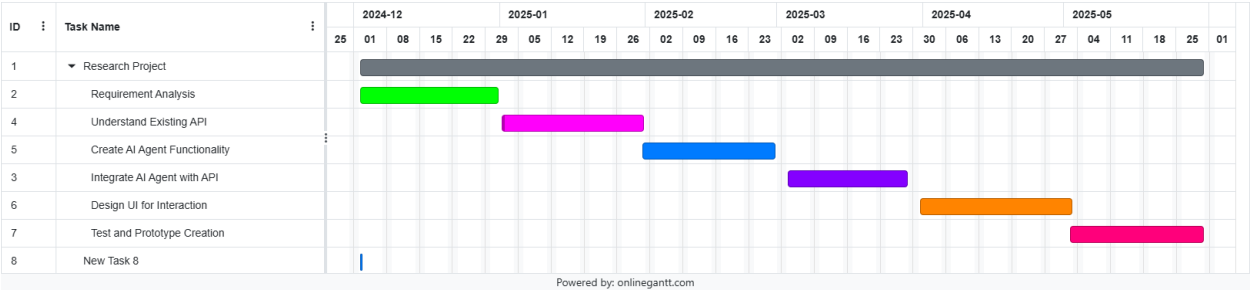

Figure 8 - Gantt Chart