Project Report On

KnowledgeNet - An Agentic framework for Deep Research

Submitted By

Soham Waghmare 22111255
Mihir Patwardhan 22111346
Riya Agarwal 22110952
Sahil Sonawane 22110784

Under the Guidance of Prof Vivek Patil

In Partial fulfilment of
Bachelor of Technology
[Artificial Intelligence and Data Science]
[2024-2025]
At

Department of Artificial Intelligence and Data Science
Affiliated To

Savitribai Phule Pune University, Pune

Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology, Pune 411048

Certificate

This is to certify that the project report entitles

“KnowledgeNet – Agentic Framework for Multimodal Deep Research”

Submitted by

Soham Waghmare	Exam No : 22111255
Mihir Patwardhan	Exam No : 22111346
Riya Agarwal	Exam No : 22110952
Sahil Sonawane	Exam No : 22110748

is a bonafide work in partial fulfillment of the award of Bachelor of Technology in Artificial Intelligence and Data Science, Savitribai Phule Pune University, Pune, during the year 2023-24. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Technology Degree.

Prof. Vivek Patil
Project Guide

Prof. Santosh Kumar
Head of Department

Prof.(Dr.)Vivek. Deshpande

Director, VIIT, Pune

Date:

External Examiner: 1. Bhushan Gawale
Examiners: 1. Prranjali Jadhav
Examiners: 2. Prasad Chaudhari

Place: Pune

Second Floor Ratnatara Complex,
Kranti Chowk, Navi Sangvi,
Pimpri Chinchwad, MH 411 027

Tel. No.: (+91) 9960163010
Web: https://mituresearch.in
Mail: contact@mitu.co.in

Date: 30.04.2025

To,
The Head-AI&DS,
VIIT, Pune

We MITU Research, Pune, hereby certify that the following students of BTech-AI&DS, VIIT have completed their final year BTech project titled: "KnowledgeNet – An Agentic Framework for Deep Research" at our central development office, Pune in the academic year 2024-2025. Team members' names with GR numbers:

Sr. No.	Name	Gr. Number
1.	Soham Waghmare	22111255
2.	Mihir Patwardhan	22111346
3.	Riya Agarwal	22110952
4.	Sahil Sonawane	22110784

We thank them for the contribution and wish them good luck for the future.

Project Director,
MITU Research, Pune

MITU RESEARCH ★ Innovative AI Engineers ★

Acknowledgement

We take this opportunity to thank Head of the Department **Prof. Dr. Santosh Kumar** and our project guide **Prof. Dr. Vivek Patil** for their valuable guidance and providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Artificial Intelligence and Data Science, VIIT, Pune for their valuable time, support, comments, suggestions and persuasion. We would also like to thank **Mrs. Prranjali Jadhav, Mr. Prasad Chaudhari** and the institute for providing the required infrastructure, internet and library facility.

Soham Waghmare	22111255
Mihir Patwardhan	22111346
Riya Agarwal	22110952
Sahil Sonawane	22110784

ABSTRACT

The KnowledgeNet framework is a multi-agent AI system designed to assist with academic research by autonomously exploring information, managing knowledge, and producing written reports. It employs distinct agents (a ResearchManager, multiple Researchers, and a ReportWriter) that collaborate under the coordination of a FastAPI-based backend. The system uses advanced web crawling (via Crawl4AI and Playwright) to gather data, stores knowledge in scalable NoSQL databases (MongoDB or DynamoDB), and provides real-time feedback through a WebSocket-powered frontend. In implementation, KnowledgeNet creates tree-structured research plans, scrapes relevant web content (preferring Google with DuckDuckGo as a fallback), manages contextual memory, and dynamically generates a formatted report. Deployed on a GCP VM (backend with Uvicorn+Nginx) and a Vercel-hosted frontend, the system demonstrates robust performance in automated knowledge acquisition and documentation. This report details the architecture, implementation, results, and future directions of the KnowledgeNet project.

Index

Chapter 1

Introduction

1.1 Motivation

The primary motivation behind the KnowledgeNet framework stems from the increasing need to streamline and enhance the process of academic research in an era of overwhelming information. With vast amounts of knowledge distributed across the internet, researchers often face challenges in locating, organizing, and synthesizing relevant data efficiently. The manual effort required to conduct thorough literature reviews or compile research findings not only consumes significant time but is also prone to human biases and oversight. KnowledgeNet aims to address these limitations by leveraging the capabilities of artificial intelligence, particularly multi-agent systems and LLMs, to automate the entire research pipeline. Inspired by the vision of making knowledge "irrepressible," the framework emphasizes structured, scalable, and context-aware exploration of information. By integrating state-of-the-art tools like Crawl4AI for web scraping and combining them with intelligent task delegation and report generation, KnowledgeNet aspires to transform research from a labor-intensive activity into an optimized and accessible process. Furthermore, it seeks to empower users—whether students, researchers, or professionals—with a tool that not only accelerates knowledge discovery but also maintains high standards of accuracy and comprehensiveness. In doing so, the project aligns with the broader goal of democratizing information and enabling faster innovation in academic and professional domains.

1.2 Need of KnowledgeNet

The KnowledgeNet framework addresses a critical need in the modern research landscape, where the vast expanse of online information often becomes a double-edged sword. While the internet has democratized access to knowledge, the sheer volume of unstructured data poses challenges in efficiently extracting, synthesizing, and presenting insights. Traditional methods of research, reliant on manual searching, data collection, and analysis, are increasingly inadequate for meeting the demands of academia, industry, and journalism, where speed, accuracy, and comprehensiveness are paramount.

Researchers often struggle with scattered resources, fragmented data, and information overload, leading to inefficiencies and missed opportunities for innovation.KnowledgeNet is designed to bridge this gap by automating the research process through an intelligent multi-agent system that mimics human research behavior. By employing tools like Crawl4AI for structured web scraping, intelligent memory management for data retention, and dynamic context handling for iterative refinement, the framework ensures a streamlined, scalable, and robust approach to information gathering. It caters to the need for real-time progress tracking, structured outputs, and multi-modal integration of data, which are essential for creating high-quality research reports. Furthermore, the need for democratized access to well-structured knowledge, particularly in under-resourced settings, makes this framework an indispensable tool for empowering individuals and institutions to achieve their research objectives more effectively. In a world where knowledge fuels progress, KnowledgeNet stands as a crucial innovation for accelerating the journey from information discovery to actionable insights.

1.3 Brief Introduction to Accommodation Management System

The contemporary landscape of information is characterized by an unprecedented rate of data generation and dissemination, posing significant challenges for researchers across diverse fields who strive to remain abreast of the latest developments and synthesize relevant knowledge. The ability to efficiently navigate, extract, and synthesize information from this vast ocean of data has become increasingly critical for advancing scholarly inquiry, informing journalistic narratives, and driving strategic business decisions. In response to this growing need, the field of research automation has emerged, leveraging the power of artificial intelligence to streamline and accelerate the research process. This report introduces the KnowledgeNet framework, an innovative AI-driven system designed to significantly condense the time required for comprehensive research, transforming what traditionally takes days into a matter of minutes. By automating key stages of the research workflow, KnowledgeNet holds the potential to enhance research efficiency and provide high-quality, synthesized content across academia, journalism, and business analysis, thereby empowering professionals in these domains to focus on higher-level analysis and decision-making.

Chapter 2

Literature Survey

1.1 Literature Review

The emerging field of AI-driven research automation is witnessing significant advancements with systems like the KnowledgeNet framework. This literature review examines key technologies and parallel approaches that contextualize KnowledgeNet's architecture and functionality. Our findings indicate that modern web crawling technologies like Crawl4AI provide essential high-speed data collection capabilities, while systems such as Elicit and LitLLM demonstrate growing maturity in automating literature synthesis tasks. The backend architecture combining FastAPI, Uvicorn, and Nginx represents current best practices for building scalable AI systems. Advancements in AI agent memory concepts and real-time visualization techniques further support KnowledgeNet's approach to condensing extensive research processes into minutes while maintaining quality and comprehensiveness.

Web Crawling and Data Collection Technologies

At the heart of research automation systems lies efficient web crawling and data extraction. The KnowledgeNet framework's use of Crawl4AI aligns with leading approaches in this domain.

Crawl4AI: High-Performance Web Crawling

Crawl4AI represents a significant advancement in web crawling technology specifically designed for AI applications. As the "#1 trending GitHub repository," Crawl4AI delivers "blazing-fast, AI-ready web crawling tailored for large language models, AI agents, and data pipelines". Its open-source nature and active community maintenance align with KnowledgeNet's need for flexible, high-performance data collection.

The technology stands out for several capabilities that make it particularly suitable for research automation:

1. **Clean Markdown Generation**: Crawl4AI excels at producing well-formatted markdown content that integrates seamlessly with RAG (Retrieval-Augmented Generation) pipelines or direct LLM ingestion. This capability directly supports

KnowledgeNet's node-based information storage approach.

2. **Advanced Browser Control**: The system offers "hooks, proxies, stealth modes, session re-use" and other fine-grained control mechanisms, enabling KnowledgeNet to navigate complex web environments and access diverse information sources.
3. **Parallel Processing**: Crawl4AI's support for "parallel crawling, chunk-based extraction" facilitates the rapid data collection needed to condense "48 hours of human research into minutes" as specified in KnowledgeNet's goals.
4. **LLM-Friendly Output**: By generating "minimally processed, well-structured text, images, and metadata," Crawl4AI produces content that AI models can easily consume, supporting KnowledgeNet's integration with language models like Gemini.

Similar AI Research Automation Systems

Several existing systems share KnowledgeNet's goal of automating research processes, providing valuable context for understanding its approach and potential advantages.

Elicit: Automated Scientific Literature Review

Elicit focuses specifically on automating scientific literature review, addressing a subset of KnowledgeNet's broader research automation capabilities. Launched in 2023, Elicit "aims to develop a tool to automate the more tedious parts of the scientific literature review process".

The system leverages "a variety of models both first- and third-party" to search and discover concepts across academic papers. Users can pose questions like "What are all of the effects of creatine?" to receive compiled answers from academic literature. While KnowledgeNet employs a tree-like structure for iterative expansion, Elicit's approach demonstrates the value of targeted question-answering for research synthesis.

According to its creators, Elicit delivers "cost and time savings to the academic and industry research organizations" and can potentially enable "just-in-time updates when the state of knowledge in a field changes". This aligns with KnowledgeNet's goal of significantly reducing research time.

LitLLM: LLM-Powered Literature Review Toolkit

LitLLM represents another specialized approach to research automation, with a focus on

scientific literature reviews. Similar to KnowledgeNet, it employs a sophisticated technical architecture to overcome limitations in existing automated literature review systems.

The toolkit "operates on Retrieval Augmented Generation (RAG) principles, specialized prompting and instructing techniques with the help of LLMs". This reflects KnowledgeNet's use of LLMs for processing and synthesizing information. LitLLM's approach includes:

1. Web search initiation "to retrieve relevant papers by summarizing user-provided abstracts into keywords"
2. Re-ranking of retrieved papers based on user-provided abstracts
3. Generation of related work sections based on the re-ranked results

While LitLLM focuses specifically on academic literature review rather than general research automation, its workflow demonstrates a proven approach to breaking down complex research tasks into manageable components-a principle also employed by KnowledgeNet.

AI Agent Memory and Management

KnowledgeNet's central memory system managed by a research manager agent represents a key component of its architecture. Understanding current approaches to AI agent memory provides context for this design choice.

Agent Memory Concepts and Implementations

AI agent memory refers to "an artificial intelligence (AI) system's ability to store and recall past experiences to improve decision-making, perception and overall performance". This capability is critical for KnowledgeNet's research manager agent, which must coordinate and synthesize information across multiple research paths.

Current literature distinguishes between different types of agent memory, categorized similarly to human memory. According to research from Princeton University, these include:

1. **Short-term memory (STM)**: Enables "an AI agent to remember recent inputs for immediate decision-making" and is "useful in conversational AI, where maintaining context across multiple exchanges is required". In KnowledgeNet, this likely facilitates

the coordination of active research paths.

While traditional AI models "process each task independently, AI agents with memory can retain context, recognize patterns over time and adapt based on past interactions". This capability is essential for KnowledgeNet's iterative research approach, where findings from one branch of investigation may inform others.

One significant challenge in implementing AI memory systems is "optimizing retrieval efficiency, as storing excessive data can lead to slower response times". KnowledgeNet's tree-like structure and centralized memory management likely address this challenge by organizing information hierarchically and maintaining relevant contextual relationships.

Backend Technologies for Scalable AI Systems

KnowledgeNet's backend architecture combines FastAPI, Uvicorn, and Nginx-a stack optimized for high-performance, scalable AI applications. Current literature provides insights into the advantages of this technical approach.

FastAPI for Microservices Architecture

FastAPI has emerged as a preferred framework for building scalable microservices, particularly for AI applications. It offers "high performance, usability, and robust support toward asynchronous programming" along with "built features that favour data validation, interactive documentation, and dependency injection".

For systems like KnowledgeNet that require efficient processing of large amounts of information, FastAPI provides several key advantages:

1. **High Performance**: FastAPI supports "asynchronous programming" to manage "multi-threaded workloads efficiently", enabling KnowledgeNet to handle multiple parallel research paths.
2. **Modularity**: The microservices architecture facilitated by FastAPI aligns with KnowledgeNet's modular design, which integrates various technologies including "WebSockets for real-time communication, Gemini LLM for long-context processing, and MongoDB/DynamoDB for efficient data storage."
3. **Scalability**: FastAPI's design supports building "maintainable, scalable and highly flexible applications", essential for KnowledgeNet's ability to handle varying research loads.

Uvicorn and Nginx Configuration

For production deployment of FastAPI applications like KnowledgeNet, Uvicorn serves as "a lightning-fast ASGI server built on uvloop and httptools". The typical setup involves installing Uvicorn with its standard dependencies:

$ pip install "uvicorn[standard]"

This includes "uvloop, which significantly enhances concurrency performance" to support KnowledgeNet's parallel processing requirements.

Uvicorn provides configuration options that allow for optimization based on workload:

- --host: To specify the host address (using "0.0.0.0" to make it accessible externally)[9]

- --port: To set the port number

- --workers: To "specify the number of worker processes to handle requests, which can improve performance under load"

This configuration flexibility supports KnowledgeNet's scalability needs, particularly when deployed on Google Cloud Platform as mentioned in the framework description.

Real-Time Visualization and Reporting

KnowledgeNet's frontend features "real-time progress tracking, interactive visualizations of the research tree, and a comprehensive rendered report." Current approaches to real-time dashboarding provide context for these capabilities.

Real-Time Dashboarding Technologies

Building effective real-time dashboards presents complex challenges that KnowledgeNet must address. Traditional approaches "using data warehouses or transactional databases with embedded business intelligence (BI) tools often fall short due to issues with data freshness, query latency, and a rocky user experience".

Modern solutions combine streaming data technologies with frontend frameworks like Next.js, which KnowledgeNet employs. This approach enables dashboards that "provide users with the most current data, presented with minimal delay, ensuring decisions can be made based on the freshest insights".

For reporting functionality, systems like the Iterative Report Builder demonstrate current approaches to AI-assisted report generation. This tool is "designed for comprehensive and adaptive research and report writing, utilizing multiple online sources"[7] and "employs an iterative process, where each phase of report development is informed by user feedback, new information, and continuous improvement"[7]. This approach parallels KnowledgeNet's dynamic report generation capabilities.

Trustworthy AI System Development

As an AI-driven research system, KnowledgeNet must consider trustworthiness in its design and implementation. Research from TCS provides a framework for building trustworthy AI systems that "uses methods and processes to improve the security, privacy, explainability, bias, and calibration of AI models"[1].

These considerations are particularly important for KnowledgeNet as it automates research processes that have traditionally relied on human judgment and critical thinking. Implementing appropriate testing and validation mechanisms would ensure that the system produces reliable, unbiased research results.

Conclusion and Future Directions

The literature review demonstrates that KnowledgeNet integrates several cutting-edge technologies and approaches in AI-driven research automation. Its combination of high-performance web crawling, centralized agent memory, scalable backend architecture, and real-time visualization aligns with current best practices while potentially advancing the state of the art through its integrated, tree-based approach.

Future research directions identified in the literature suggest several potential paths for KnowledgeNet's evolution. These include enhanced multimodal data integration (already mentioned as a future expansion), improved memory management systems for AI agents, and more sophisticated approaches to ensuring the trustworthiness of automated research outputs.

Chapter 3

Project Statement

3.1 Purpose behind the Project

The purpose of the KnowledgeNet framework is to revolutionize the way research is conducted by providing an automated, efficient, and reliable system for exploring, analyzing, and synthesizing information. Designed to mimic and enhance the capabilities of a human researcher, KnowledgeNet aims to bridge the gap between the vast, unstructured data available on the internet and the need for concise, structured, and actionable insights. By leveraging cutting-edge technologies such as multi-agent systems, LLMs, and advanced web scraping tools like Crawl4AI, the framework aspires to alleviate the challenges associated with traditional research methods, including time consumption, information overload, and potential human error.

3.2 Decision of Scope

The system's ability to autonomously create research plans, traverse interconnected data in a tree-like structure, and iteratively refine knowledge ensures that users receive high-quality, comprehensive reports tailored to their queries. The integration of real-time dashboards, visualized research trees, and interactive outputs further enhances its usability and accessibility, making it an invaluable tool for academics, professionals, and organizations alike. Ultimately, the purpose of KnowledgeNet is to democratize access to information, empower decision-making, and foster innovation by transforming research into an optimized, scalable, and user-centric process.

3.3 Methodology for solving this proposed theme

3.3.1 Proposed system Architecture

Architecture and Functionality

The KnowledgeNet framework represents an AI-driven research automation system engineered to significantly reduce the time and effort typically associated with extensive research endeavors. At its core, KnowledgeNet is designed to condense approximately 48 hours of traditional human research into a matter of minutes, thereby offering a

substantial enhancement in research efficiency. The framework employs a scalable architecture, featuring a robust backend built using FastAPI, hosted on Google Cloud Platform (GCP) and served by Uvicorn and Nginx. This backend infrastructure ensures the system's reliability and ability to handle varying loads. Complementing the backend is a user-friendly frontend developed with Next.js, hosted on Vercel, which provides an interactive interface for users to engage with the research process and access the generated reports. The modular design of KnowledgeNet allows for the seamless integration of several key technologies, including WebSockets for real-time communication, the Gemini LLM for advanced long-context processing, and MongoDB/DynamoDB for efficient and scalable data storage.

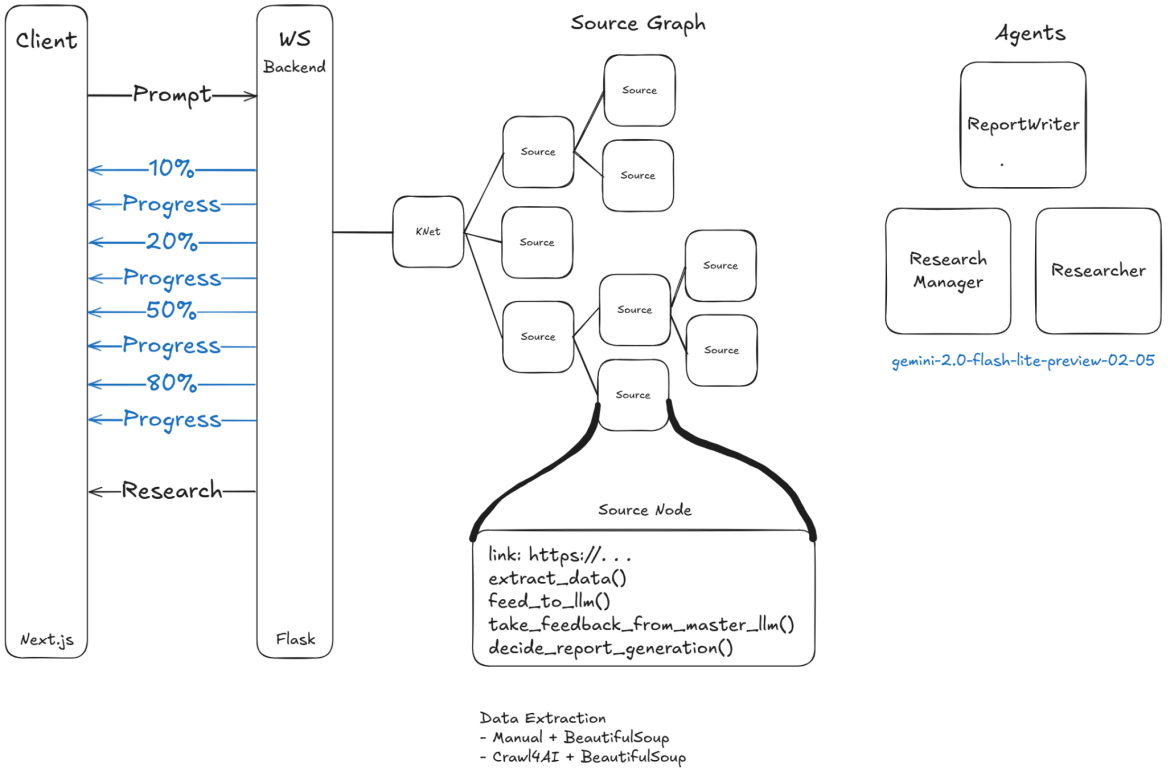


Fig 3.3.1: Proposed system Architecture

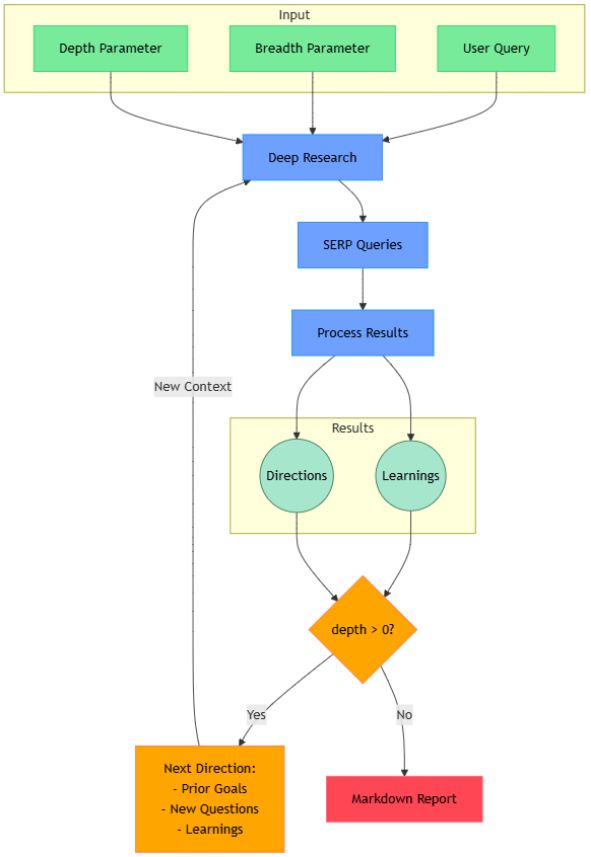


Fig 3.3.2: Working of the System

1. The research process within the KnowledgeNet framework is structured around a tree-like architecture. Users initiate the process by specifying a topic of interest, which serves as the root of the research plan. From this root, the system dynamically spawns nodes in a hierarchical structure. Each node represents a specific web search query related to the overarching research topic. The framework iteratively expands this tree by conducting web searches based on the queries defined in each node. To facilitate rapid and efficient data collection, KnowledgeNet leverages Crawl4AI, an open-source web crawler and scraper, for parallel web scraping. Crawl4AI is specifically designed for AI and large language model applications, capable of extracting clean, structured data in formats like Markdown. This parallel data collection ensures that the nodes in the research tree are populated with relevant information quickly and efficiently.
2. Within each node of the research tree, the KnowledgeNet framework stores a variety of data types. This includes markdown-formatted text extracted from relevant web pages, key images that provide visual context to the research topic, and embedded videos that offer multimedia perspectives. The content gathered in each node is then fed into a central memory system managed by a dedicated research manager agent. This central memory acts as a repository of the

information collected throughout the research process, allowing the framework to maintain a comprehensive understanding of the topic.

3. The KnowledgeNet framework utilizes the information stored in its central memory to dynamically create a report outline. This outline serves as the structural backbone for the final research report. The framework iteratively fills the sections of this outline with detailed, high-quality content derived from the data collected in the research tree. This dynamic and iterative approach ensures that the report is comprehensive, well-organized, and directly relevant to the user-specified topic. By automating the processes of web searching, data collection, content organization, and report generation, KnowledgeNet provides a powerful solution for significantly accelerating the research workflow and delivering valuable insights in a fraction of the time required by traditional methods.

Chapter 4

System requirement and specification

4.1 Software Requirements Specifications

4.1.1 Introduction

Intended Audience and Reading Suggestions:
The KnowledgeNet framework is designed for developers, researchers, students, and other users who require an automated, intelligent system for conducting in-depth research. The intended audience includes:

1. **Developers** - For implementing and maintaining the system.
2. **Users** - Researchers, students, and professionals seeking knowledge.
3. **Testers** - For ensuring system robustness.
4. **Documentation Writers** - To maintain detailed system documentation.

4.1.2 User Classes and Characteristics

There are three primary agents in the system:

1. **Users:**
    a. Submit research queries to the system.
    b. Interact with generated research reports, visualized research trees, and dashboard elements (images, graphs, links, videos).
    c. Ask follow-up questions or refine queries.
2. **ResearchManager Agent:**
    a. Creates and oversees the research plan.
    b. Manages tree-like structures for organizing nodes.
    c. Stores key findings in memory for report generation.
3. **ReportWriter Agent:**
    a. Generates a comprehensive, structured report using iterative LLM refinement.
    b. Dynamically fills outlines created by ResearchManager.

4.1.3 Operating Environment

The system is a web-based application hosted with the following setup:

- **Backend**: Deployed on GCP virtual machine with Uvicorn and Nginx.
- **Frontend**: Hosted on Vercel.
- Requires modern web browsers and a stable internet connection for seamless operation.

4.1.4 External Interface Requirements

User Interfaces:

The frontend dashboard allows:

- Submission of research queries.
- Real-time progress tracking of research nodes.
- Visualization of the research tree and generated reports.

Hardware Interfaces:

- Client device with internet access.

Software Interfaces:

- Crawl4AI for web scraping and data extraction.
- Python backend using FastAPI for task management.
- Next.js for the frontend dashboard.

4.1.5 Functional Requirements

1. **Query Submission**:
    a. Users submit research topics via a web interface.
2. **Research Tree Management**:
    a. Dynamically visualize the research tree structure in real-time.
3. **Report Generation**:
    a. Generate structured, multimodal research reports.
4. **Interactive Dashboard**:
    a. View and interact with research findings, including images, graphs, and source links.

4.1.6 Non-functional Requirements

1. **Data Input Requirements**:
    a. Efficient handling of text, images, and video data from web sources.
2. **Security Requirements**:
    a. Authentication for user accounts.
    b. Secure data handling to prevent unauthorized access.
3. **Software Quality Attributes:**
    a. **Reliability**: Accurate extraction and reporting of data.
    b. **Efficiency**: Optimize resource usage during web scraping and report generation.
    c. **Scalability**: Support for multiple concurrent queries.
    d. **Flexibility**: Modular design for incorporating new features or agents.

4.1.7 Product Perspective

KnowledgeNet is an open-source, client-server model system designed for automated research.

1. **User Account:** Allows users to track and refine research projects.
2. **Search Engine Integration:** Uses Crawl4AI for dynamic web crawling.
3. **Visualization:** Offers an interactive tree-based view of the research process.
4. **Multi-agent Collaboration:** Enables seamless interaction between ResearchManager, Researcher, and ReportWriter agents.

4.1.8 Product Function

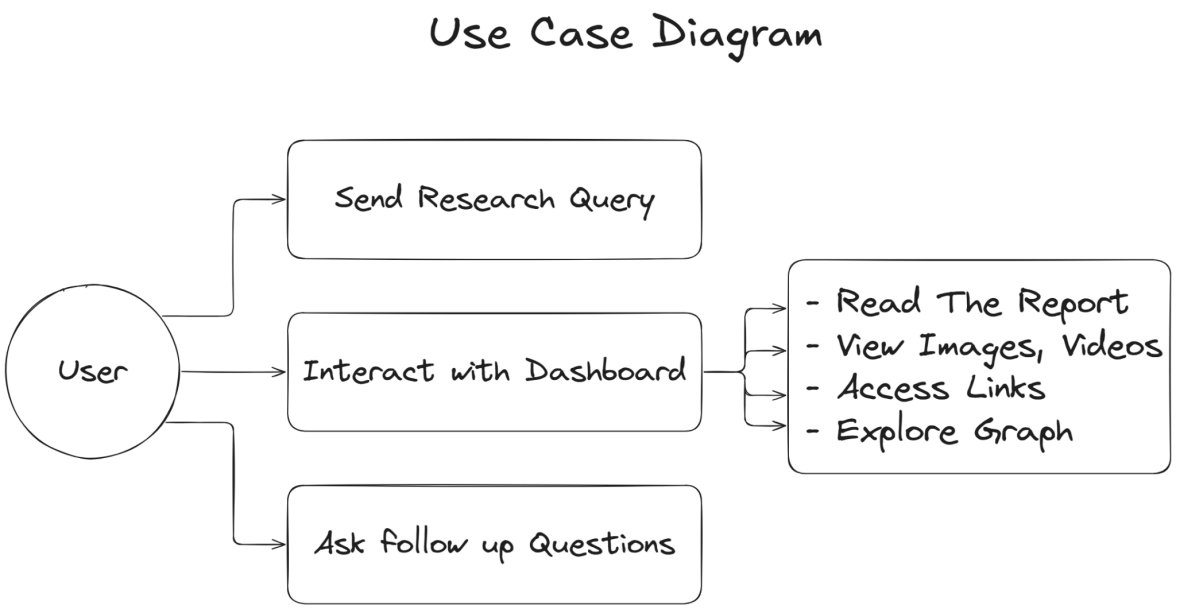
Major functionalities include:

1. **Research Management:**
    a. Organize research nodes in a hierarchical tree.
    b. Store results and sources for transparency.
2. **Dashboard Features:**
    a. Real-time updates on research status.
    b. Comprehensive visualizations of findings.
3. **Report Output:**
    a. Create and export detailed reports, including multimedia elements.
4. **System Integration:**
    a. Smooth interaction between frontend, backend, and data sources.

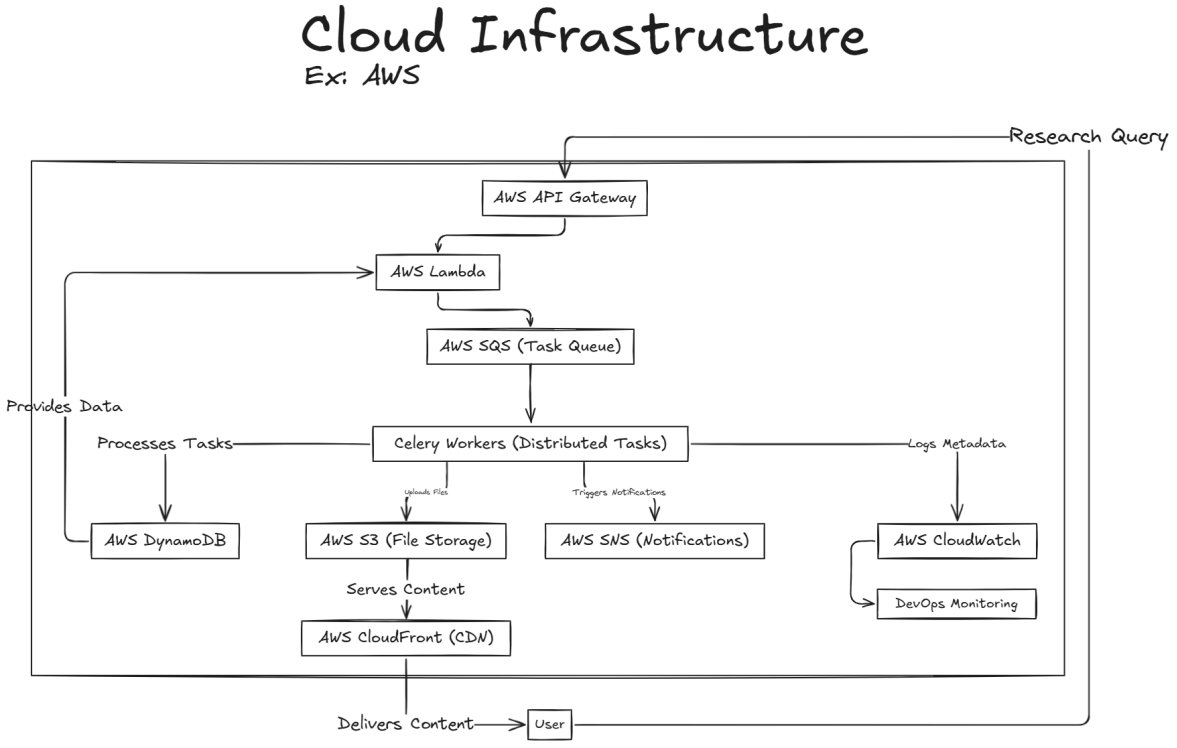
Chapter 5

Project Analysis and Design

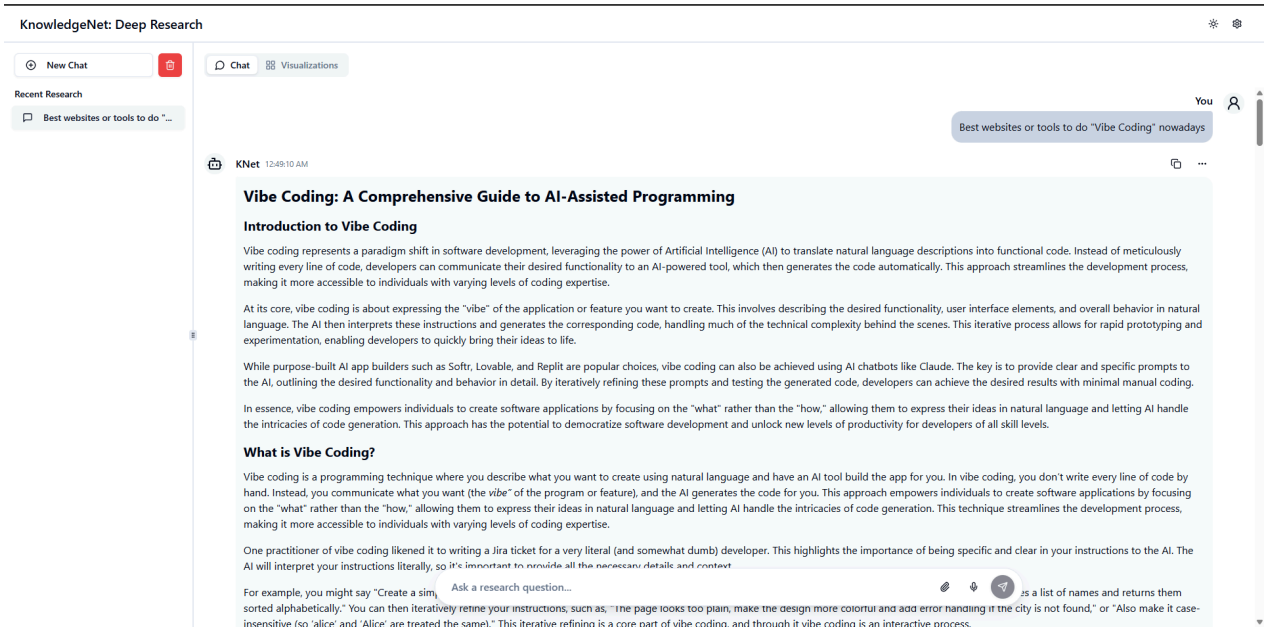
5.1 Use Case Diagram



5.2 Cloud Infrastructure Diagram



5.3 Interface and Details



Chapter 6

Implementation Software Testing

1. Introduction

This document provides a comprehensive overview of the testing strategy for the KnowledgeNet system. The project's objective is to provide a robust platform for automating and streamlining research tasks using an intelligent multi-agent system. It highlights the testing standards that will be applied at various levels, including unit, integration, and system testing, ensuring the reliability and scalability of the system.

2. Purpose

The project aims to develop a research automation framework that leverages AI-driven agents to perform research tasks such as web scraping, dynamic data processing, and report generation. The KnowledgeNet system delivers a user-friendly interface for researchers, professionals, and students to initiate and manage research processes seamlessly. The backend facilitates real-time data flow and dynamic research management, while the frontend provides a clear, interactive visualization of the research progress and outputs.

3. Test Objective

The objective of our test plan is to identify and document bugs to enhance the system's reliability, performance, and user experience. While exhaustive testing is not feasible, a broad spectrum of test scenarios will be conducted to ensure the integrity of the application. Key functionalities to be tested include:

1. **Research Query Submission**: Ensure correct initialization of the research process.
2. **Node Management**: Validate tree-structured data traversal and node management.
3. **Web Scraping**: Test the effectiveness and accuracy of the scraping mechanism.
4. **Report Generation**: Confirm that the generated reports meet expected quality and structure.
5. **Real-Time Updates**: Verify live progress tracking and research tree visualization.

The application is designed to run seamlessly across various platforms and ensure minimal impact on performance and usability.

4. Process Overview

The testing process for the KnowledgeNet system includes the following steps:

1. **Requirements Identification**:
    - Define the requirements for each component.
    - Derive test cases from the current system specifications.
2. **Module Testing**:
    - Identify the specific tests to be used for each module.
    - Review and refine test data and cases to ensure thorough coverage.
3. **Test Execution**:
    - Document test configurations, input data, and expected results.
    - Execute tests to validate system modules and their interactions.
4. **Bug Reporting**:
    - Generate bug reports for any failed tests, detailing the problem, possible

cause, and sequence of events leading to the failure.
    - Submit reports for technical analysis and resolution.
5. **Integration and System Testing**:
    - Ensure successful unit tests are completed before integrating components.
    - Conduct system-level tests to validate overall functionality.
6. **Documentation and Reporting**:
    - Submit test results and update specifications based on feedback.
    - Revise and refine the system as needed for optimal performance.

5. Test Cases and Results

Test Case 1: Research Query Submission

- Objective: Validate the submission of a research topic and initiation of research planning.
- Inputs: User-submitted query.
- Expected Result: Proper creation of a research plan with no errors.
- Outcome: Pass/Fail.

Test Case 2: Web Scraping Efficiency

- Objective: Ensure data scraping is performed accurately from multiple sources.
- Inputs: Web search strings and URLs.
- Expected Result: Retrieved data is correctly formatted in markdown, with relevant images and videos extracted.
- Outcome: Pass/Fail.

Test Case 3: Research Tree Visualization

- Objective: Test the real-time rendering of the research tree on the frontend.
- Inputs: Tree data from the backend.
- Expected Result: Accurate and responsive visualization of the tree structure.
- Outcome: Pass/Fail.

Test Case 4: Report Generation

- Objective: Verify that the system produces well-structured, detailed reports.
- Inputs: Data stored in the ResearchManager's memory.
- Expected Result: Complete and coherent report with properly organized headings and content.
- Outcome: Pass/Fail.

Test Case 5: Real-Time Updates

- Objective: Ensure smooth real-time updates for users.
- Inputs: Progress events from the backend.
- Expected Result: Real-time progress is displayed without delays.
- Outcome: Pass/Fail.

Chapter 7

Conclusion and Future work

7.1 Conclusion

The KnowledgeNet framework represents a significant advancement in the domain of AI-driven research automation. By strategically integrating a suite of cutting-edge technologies, including Crawl4AI for efficient data acquisition, a scalable backend on GCP powered by FastAPI, Uvicorn, and Nginx, an interactive frontend on Next.js and Vercel, WebSockets for real-time communication, the powerful Gemini LLM for long-context processing, and flexible data storage solutions with MongoDB and DynamoDB, the framework offers a novel approach to condensing extensive research processes into a matter of minutes. Its potential to streamline workflows and enhance productivity across academia, journalism, and business analysis positions it as a valuable tool in today's information-rich environment. As the field of AI continues to evolve, the KnowledgeNet framework serves as a compelling example of how artificial intelligence can be leveraged to augment human intellect and accelerate the crucial process of knowledge discovery and synthesis. This undergraduate project contributes meaningfully to the ongoing exploration and development of AI-driven solutions for addressing the ever-increasing demands of research in the digital age. As per comprehensive evaluation using TF-IDF with cosine similarity our method is comparable to similar workflows provided by OpenAI, Google, Perplexity and Grok.

ID	Sources Used for Similarity	Average Similarity	Max Similarity
1	15	0.2853	0.6211
2	22	0.1976	0.4532
3	12	0.3421	0.7150
4	18	0.2205	0.5008
5	25	0.1750	0.3987

Table: Similarity scores across multiple samples

7.2 Future Scope

The KnowledgeNet framework has identified several potential future expansions. One key target is enhancing the system's scalability through the implementation of containerized workers. Containerization technologies like Docker can provide a more efficient and manageable way to scale the computational resources required by the framework, allowing it to handle larger research tasks and a greater number of concurrent

users. Another future direction involves broadening the AI-assisted decision-making capabilities of the framework. While currently focused on research automation, future iterations could potentially expand to assist with more complex decision-making processes across academia, journalism, and business analysis, leveraging the insights derived from the automated research.

References

Web Crawling & AI Automation

[1] Huang, C-J. (2024). *The Synergy of Automated Pipelines with Prompt Engineering and Generative AI in Web Crawling*. arXiv:2502.15691.

[2] Mesquita, F., et al. (2019). *KnowledgeNet: A Benchmark Dataset for Knowledge Base Population*. ACL Anthology.

[3] K.Rekha, Kailash Kumar (2024). *Survey Significance of AI for Ensuring Precise Web Crawling*. IJCRT.

[4] Collins, C., et al. (2021). *Artificial Intelligence in Information Systems Research: A Systematic Review*. ScienceDirect.

[5] Dennehy, D., et al. (2021). *AI-Driven Automation in Organizational Processes*. ScienceDirect.

AI Memory & Agent Architectures

[6] Chuanyang, H., & He, Q. (2025). *Enhancing Memory Retrieval in Generative Agents through LLM-Trained Cross Attention Networks*. Frontiers in Psychology.

[7] Barbosa, D., et al. (2019). *Knowledge Base Population via Hybrid NLP Pipelines*. ACL Anthology.

[8] Mikalef, P., et al. (2021). *AI Agent Memory Systems for Decision-Making*. ScienceDirect.

AI Efficiency & Research Automation

[9] Author Anonymous (2024). *The Impact of Artificial Intelligence on Research Efficiency*. ScienceDirect.

[10] He, Q., et al. (2025). *LLM-Assisted Memory Optimization for Multi-Agent Systems*. Frontiers in Psychology.

[11] DigitalOcean (2024). *AI Tools for Streamlined Knowledge Exploration*. DigitalOcean Research.

[12] Scite.ai (2024). *Collaborative AI Platforms for Research Synthesis*. DigitalOcean.

[13] Elicit.org (2024). *Automated Systematic Reviews Using NLP*. DigitalOcean.

Software Engineering & AI Integration

[14] Author Anonymous (2025). *AI-Driven Innovations in Software Engineering*. MDPI Applied Sciences.

[15] FastAPI Documentation (2023). *Scalable Microservices for AI Systems*. FastAPI Official Resources.

[16] Uvicorn Contributors (2023). *High-Performance ASGI Server Design*. Uvicorn GitHub Repository.

Ethics & Human-AI Collaboration

[17] Brynjolfsson, E., et al. (2023). *The Blended Future of Automation and AI*. ScienceDirect.

[18] Conboy, K., et al. (2021). *Ethical AI in Organizational Decision-Making*. ScienceDirect.

[19] Rekha, K. (2024). *Moral Considerations in AI-Powered Web Crawling*. IJCRT.

[20] Chuanyang, H. (2025). *Bias Mitigation in Generative Agent Memory*. Frontiers in Psychology.