Back to publications

Mar 11, 2025 • 33 reads • MIT License

Nexus: A Three-Pronged Agentic AI System for Intelligent Research and Analysis

3-Agent System Agentic ai AI Gemini LangChain LangGraph LLM Multi-Agentic AI Researcher model Serper API

Mohit Kumar D Diya Ravishankar

Table of contents

Abstract

Nexus is an advanced AI-powered research system designed to automate and enhance the process of gathering, analyzing, and generating structured research content. Traditional research involves **time-consuming manual searches, information filtering, and content structuring**, which can introduce inefficiencies and inconsistencies. Nexus overcomes these challenges by utilizing a **multi-agent AI architecture**, integrating **natural language processing (NLP), web scraping, and AI-driven content synthesis** to streamline research workflows.

Nexus employs **three specialized AI agents**:

1. **Research Agent** – Collects real-time research data from trusted sources such as **Google Search, Wikipedia, and ArXiv**.
2. **Reporting Agent** – Processes collected data and structures it into a **coherent, well-formatted research article**.
3. **Storage Agent** – Saves research outputs in **Markdown (`.md`), JSON (`.json`), and PDF formats** for accessibility and further usage.

Key features of Nexus include **customizable tone settings**, a **fact-checking mechanism using Google's Fact Check API**, and **seamless user interaction via a frontend interface built with React and Tailwind CSS**. With **automation, AI integration, and structured workflows**, Nexus significantly reduces research time, enhances accuracy, and ensures high-quality content generation.

Link to Project:

Nexus

1. Introduction

Research is essential in academic, business, and technological domains, yet traditional research methods often require **significant manual effort and verification**. With the vast amount of online information, researchers face challenges such as **misinformation, scattered data sources, and the need for structured content**.

Objectives of Nexus

- **Automate research workflows** to reduce time and effort.
- **Generate structured research articles** with a logical flow.
- **Verify content accuracy** using fact-checking mechanisms.

- **Allow tone customization** for research reports.
- **Enable multiple output formats** (`.md` , `.json` , `.pdf`).

Nexus eliminates **information overload and research inefficiencies** by using AI-powered automation.

2. System Architecture

Nexus follows a **modular AI-driven agentic architecture**, where three specialized AI agents work collaboratively.

2.1 Backend (Flask & LangGraph)

- **Handles HTTP requests** from the frontend.
- **Manages agent execution** and research processing.
- **Uses Flask** for API development.

2.2 Agents

- **Research Agent**: Extracts raw data from multiple sources.
- **Reporting Agent**: Structures the collected data into a **technical article**.
- **Storage Agent**: Saves generated content and allows **PDF exports**.

2.3 Tools & APIs

- **Google Search API** – Fetches relevant research sources.
- **Wikipedia API** – Retrieves summarized encyclopedic content.
- **ArXiv API** – Extracts academic papers.
- **Google Fact Check API** – Ensures accuracy.

- **LangGraph** – Manages AI agent workflows.
- **WeasyPrint** – Converts research output to **PDF**.
- **Flask-CORS** – Enables **frontend-backend communication**.

2.4 Frontend (React & Tailwind CSS)

- **Displays research results** in an interactive UI.
- **Provides a settings page** for tone customization.
- **Uses Framer Motion** for animations.

2.5 Storage System

- **Stores research outputs** in `.md` and `.json` formats.
- **Generates PDFs** for downloadable reports.

3. Backend Components and Code Breakdown

3.1 Main Workflow (`main.py`)

The backend initializes the **workflow pipeline**, ensuring that the three AI agents execute sequentially.

```
from langgraph.graph.state import StateGraph
from config import generate_response
from state import ResearchState
from workflow_nodes import get_research, generate_news_article, save_output

def create_workflow():
    workflow = StateGraph(state_schema=ResearchState)

    workflow.add_node("research_agent", get_research)
    workflow.add_node("reporting_agent", generate_news_article)
    workflow.add_node("storage_agent", save_output)

    workflow.add_edge("research_agent", "reporting_agent")
    workflow.add_edge("reporting_agent", "storage_agent")

    workflow.set_entry_point("research_agent")
    workflow.set_finish_point("storage_agent")

    return workflow.compile()
```

This function defines the workflow:

Research Agent → Extracts research data.
Reporting Agent → Generates a structured article.
Storage Agent → Saves the article and allows PDF export.

4. Detailed Breakdown of Each Agent

4.1 Research Agent (Data Collection)

The Research Agent collects data from multiple sources and fact-checks the results.

```
def get_research(state: dict) -> dict:
    research_tool = ResearchTool()
    topic = state["topic"]

    google_results = research_tool.search_google(topic)
    wiki_results = research_tool.search_wikipedia(topic)
    arxiv_results = research_tool.search_arxiv(topic)
    fact_check = research_tool.fact_check(topic)

    research_summary = f"Google: {google_results}\n\nWikipedia: {wiki_results}\n\nArxiv: {arxiv_results}"

    return {"research_summary": research_summary, "fact_check_results": fact_check}
```

4.2 Reporting Agent (Content Generation)

The Reporting Agent converts research data into a structured article.

```
def generate_news_article(state: dict) -> dict:
    research_state = state

    if not research_state["research_summary"].strip():
        research_state["article"] = "No research data available."
        return research_state

    article_prompt = f"""
    Based on the following research, write a compelling and detailed news article:

    {research_state["research_summary"]}

    **Ensure the article follows this format:**
    - **Title**
    - **Introduction**
    - **Key Insights**
    - **Industry Impact**
    - **Future Prospects**
    - **Conclusion**
    """

    response = generate_response(article_prompt)
    research_state["article"] = response.strip()

    return research_state
```

4.3 Storage Agent (Data Persistence)

The Storage Agent saves research results in `.md` and `.json` formats.

```
def save_output(state: ResearchState) -> dict:
    topic = state["topic"].replace(" ", "_")
    article = state["article"]
    research_summary = state["research_summary"]

    if not article.strip():
        state["message"] = "⚠️ No article content to save."
        return state

    article_path = os.path.join(BASE_DIR, f"{topic}.md")
    json_path = os.path.join(BASE_DIR, f"{topic}.json")

    os.makedirs(BASE_DIR, exist_ok=True)

    with open(article_path, "w", encoding="utf-8") as f:
        f.write(f"# {topic}\n\n{article}")

    return state
```

5. Results

Nexus successfully streamlines the research process through automation, AI-driven content generation, and structured workflows. Below are the key outcomes observed:

5.1 Functionality Demonstration

- **Automated Research:** Extracts **real-time** research data from **Google, Wikipedia, and ArXiv**.
- **AI-Generated Content:** Produces **coherent, structured** articles.
- **Customizable Tone:** Allows users to select the **tone** of the article (formal, technical, simplified, etc.).
- **Multi-Format Storage:** Saves research output in **Markdown (`.md`), JSON (`.json`), and PDF**.
- **Fact-Checking:** Verifies accuracy using **Google Fact Check API**.

- **Interactive UI:** Displays results in an **intuitive frontend interface**.
- **PDF Generation:** Provides **downloadable** research reports.

5.2 Example Output

Topic: ***Impact of AI on Healthcare***

Generated Article:

1. **Introduction**
- Overview of AI in healthcare.

2. **AI-Driven Diagnostics**
- How AI assists in disease detection.

3. **Future Trends**
- The future role of AI in medicine.

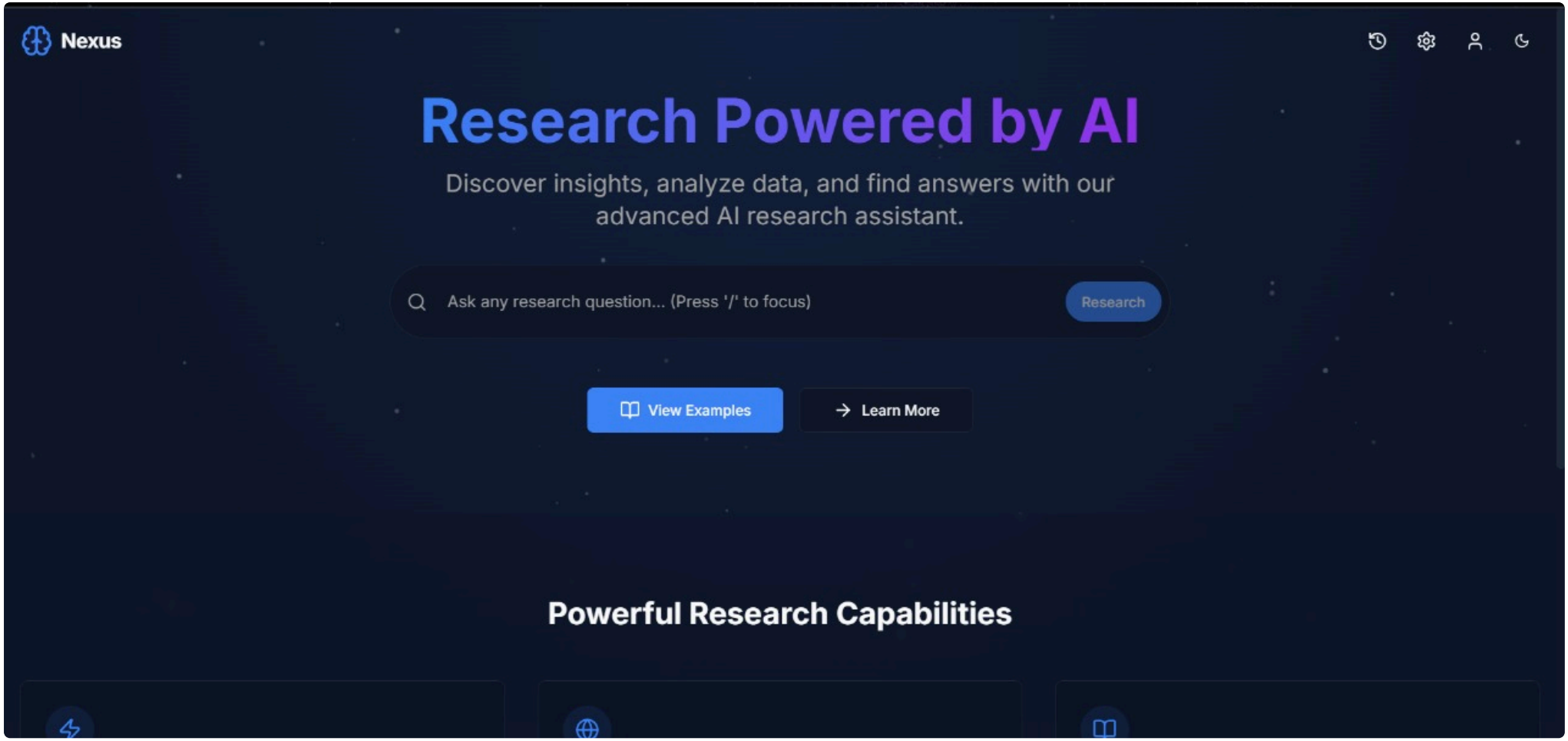
4. **Conclusion**
- Summary and impact.

6. UI/UX (Frontend)

Nexus features a **modern, responsive UI** that allows users to input a topic, process research, and view/download results. Below are images illustrating different sections of the UI.

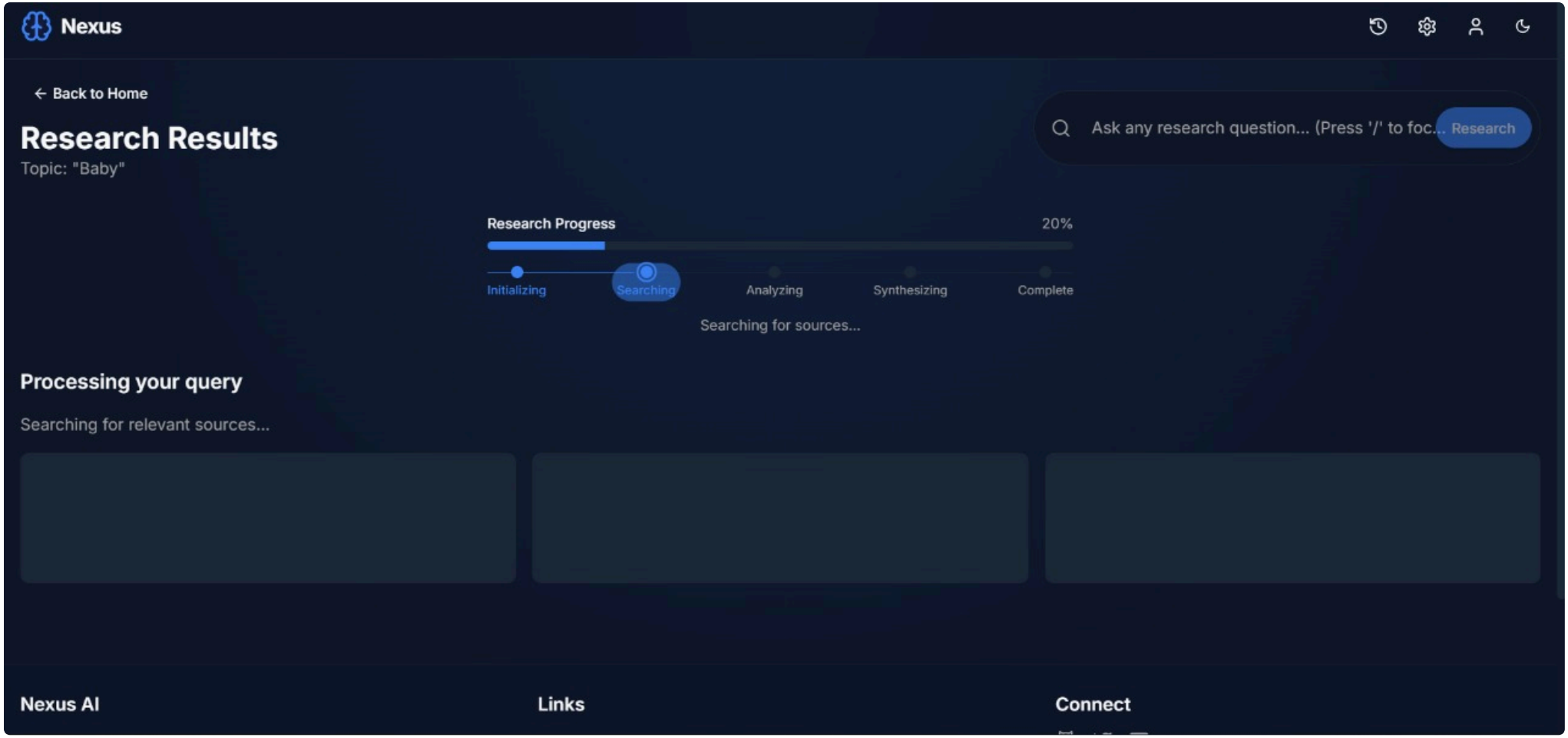
6.1 Home Page

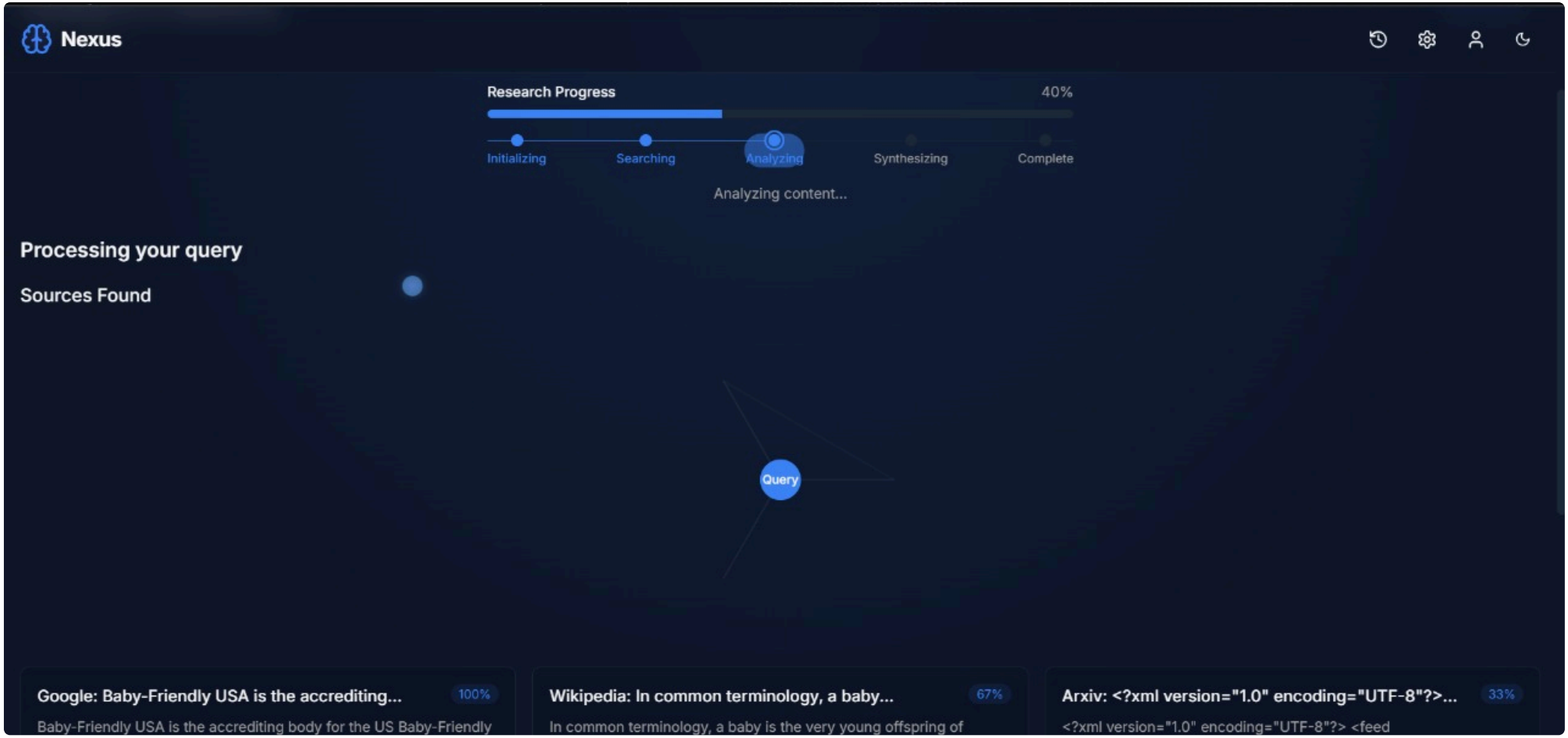
Description: The homepage where users can enter a research topic.

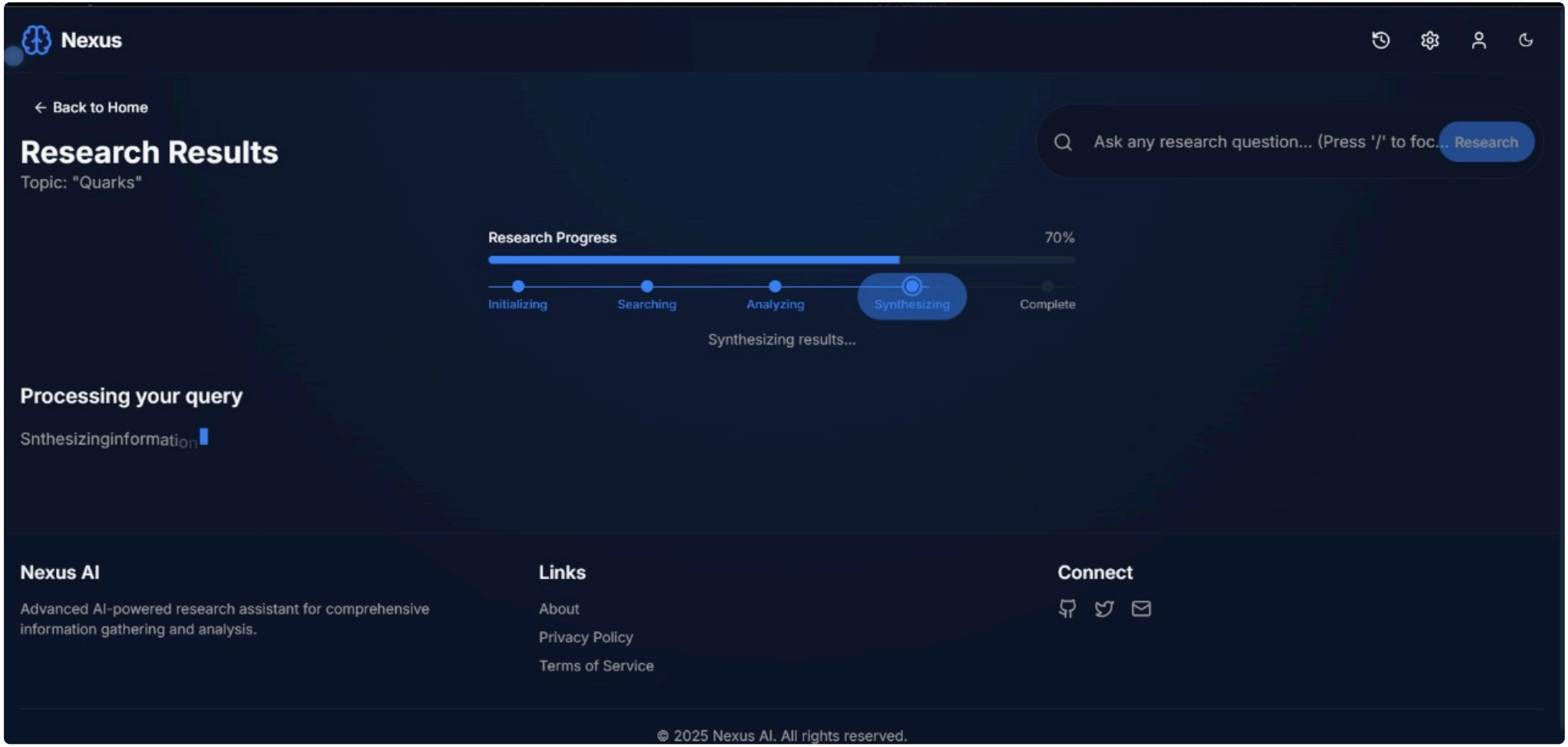


6.2 Research Processing

Description: The system processing the research request in real-time.

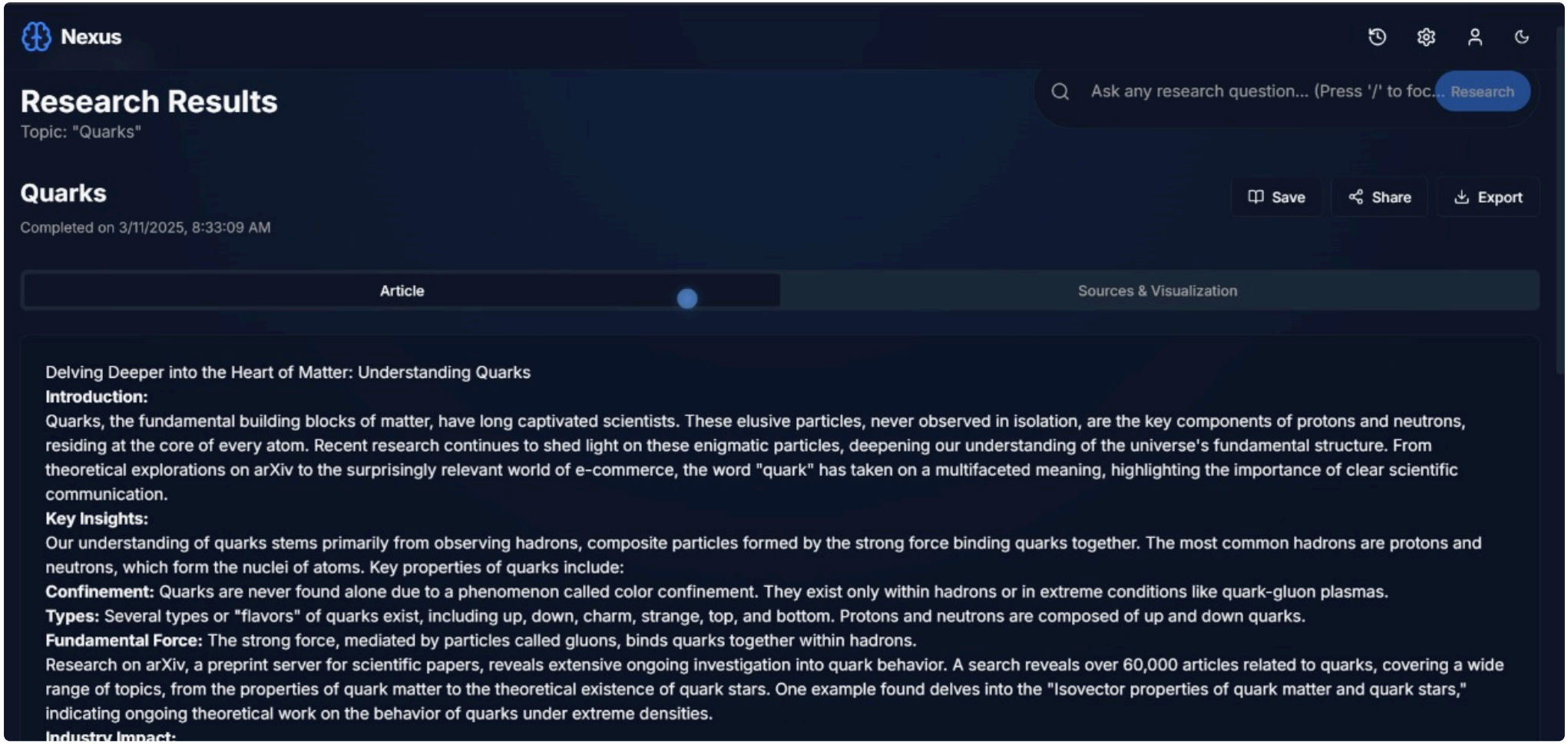

Nexus
← Back to Home
Research Results
Topic: "Baby"
Ask any research question... (Press '/' to foc...
Research
Research Progress
20%
Initializing
Searching
Analyzing
Synthesizing
Complete
Searching for sources...
Processing your query
Searching for relevant sources...
Nexus AI
Links
Connect


Nexus
Research Progress
40%
Initializing
Searching
Analyzing
Synthesizing
Complete
Analyzing content...
Processing your query
Sources Found
Query
Google: Baby-Friendly USA is the accrediting...
100%
Baby-Friendly USA is the accrediting body for the US Baby-Friendly
Wikipedia: In common terminology, a baby...
67%
In common terminology, a baby is the very young offspring of
Arxiv: <?xml version="1.0" encoding="UTF-8"?>...
33%
<?xml version="1.0" encoding="UTF-8"?> <feed



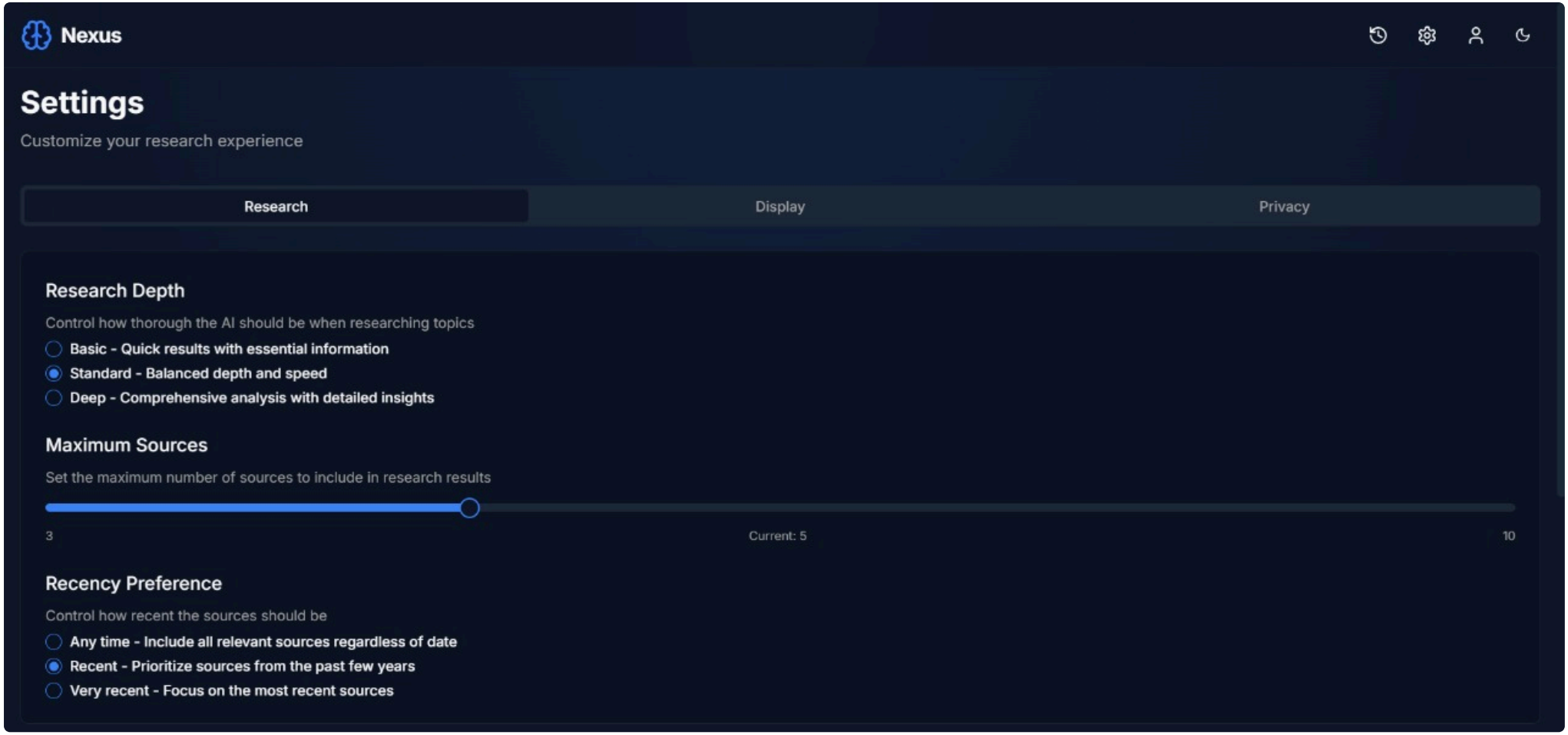
6.3 Results Display

Description: A detailed article is displayed with structured content.



6.4 Settings Page

Description: Users can adjust the tone of the research article.



7. Conclusion

Nexus provides an **AI-powered solution** for **efficient, accurate, and structured research**. By automating information retrieval, content synthesis, and storage, Nexus eliminates the manual workload traditionally associated with research.

7.1 Key Takeaways

- **AI-driven research automation** significantly **reduces research time**.

- **Structured article generation** ensures **clarity and readability**.
- **Fact-checking integration** enhances **credibility**.
- **Multi-format storage options** make research results more **accessible**.

7.2 Future Enhancements

- **Cloud-based storage** for persistent access to research history.
- **Integration with additional research sources** for broader knowledge coverage.
- **Support for multiple languages** for global accessibility.
- **Advanced AI models** for more **nuanced content generation**.

8. Tech Stack

Nexus is built using **modern web technologies** to ensure efficiency, scalability, and a seamless user experience.

8.1 Frontend

- **React.js** → Frontend framework for UI development.
- **Tailwind CSS** → Utility-first styling for a modern, responsive interface.
- **Framer Motion** → Smooth animations and UI interactions.

8.2 Backend

- **Flask** → Python-based backend framework for API handling.
- **LangGraph** → Manages agent workflow execution.
- **WeasyPrint** → Converts research output to **PDF**.

8.3 APIs & AI Models

- **Google Search API** → Retrieves online research data.
- **Wikipedia API** → Fetches encyclopedic knowledge.
- **ArXiv API** → Extracts academic research papers.
- **Google Fact Check API** → Verifies claim authenticity.
- **Google Gemini API** → AI-driven **text generation** for structured content.

Comments

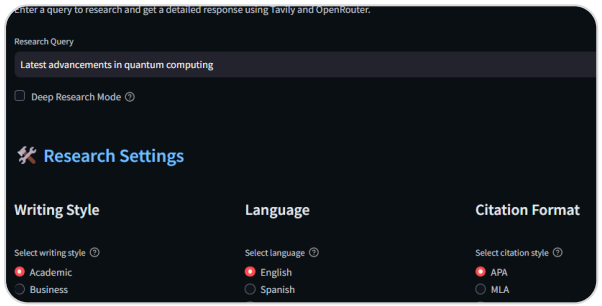
Files

example_output.pdf

No comments yet!

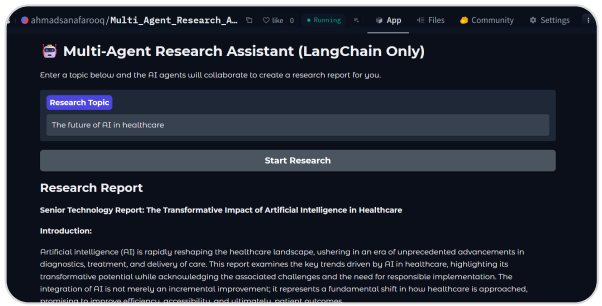
Sign in or create an account to comment

You might be interested



AI Agent Based Deep Research

AI Research Agent



Multi-Agent Research Assistant

1

Mar 22, 2025 • 20 reads

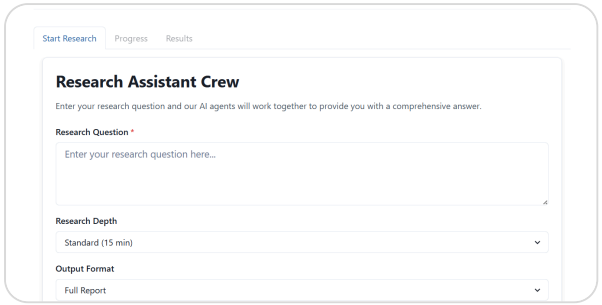
ai-research document-generation +11

May 30, 2025 • 16 reads

Python - Google API - Web Scrape

Jul 31, 2025 • 11 reads

Langchain MULTI-AGENT +2



Cerebral Collective: A Multi-Agent Research Intelligence Platform

Z

Jul 13, 2025 • 62 reads

academic-research ai-orchestration

+10