Hackathon Report: Enhancing OpenDeepSearch with Tool-Augmented LLMs

During the hackathon, our team worked on extending the capabilities of the OpenDeepSearch framework by integrating large language models and external reasoning tools. Our primary goal was to improve the factual reliability, reasoning depth, and flexibility of the system for complex question-answering tasks.

To begin with, we replaced the default model with **LLaMA 70B**, which served as the main language model for all LLM-related tasks.

We then added two key tools to augment the model's reasoning capabilities. The first was a **SPARQL Wikipedia tool**, which allows the model to issue structured queries against Wikidata. This tool enables the system to retrieve accurate factual information and perform multi-step reasoning using semantic knowledge graphs. The second addition was a **code interpreter**, referred to internally as the **fact-checker**. This module enables the LLM to verify claims, compute values, and evaluate logical conditions using Python code execution, offering an additional layer of factual accuracy and interpretability.

Alongside these architectural improvements, we also introduced a set of **new prompts** specifically designed to guide the model in using these tools effectively. These prompts focus on structured reasoning, fact-checking, and answer justification, and were tested on datasets such as FRAMES and simple_qa.

Looking ahead, we identified several promising directions for further development. These include **integrating more smolagents** to support a wider range of tasks, **improving the performance and reliability of the WikiAgent**, and **expanding the prompt library** to improve the model's reasoning behavior across different scenarios. Finally, maintaining a collaborative and exploratory atmosphere — or simply put, making space to **vibe** — was an important part of our hackathon experience and contributed greatly to our progress.