



# Stellar Mind AI

by Stellar Minds

The challenge:

**Build a Space Biology  
Knowledge Engine**

The team:

- **Candela Dávila Moreno**  
Head of Design
- **Eulogio Quemada Torres**  
CTO
- **Javier Montes Pérez**  
Head of AI
- **Alejandro Román Sánchez**  
Head of UX
- **Macorís Decena Giménez**  
CPO, Scrum Master





## INDEX

<b>1. Executive Summary .....</b>	<b>3</b>
<b>2. Context and Motivation .....</b>	<b>4</b>
<b>3. Project Goals .....</b>	<b>4</b>
General objective.....	4
Specific objectives.....	5
<b>4. Work Methodology .....</b>	<b>5</b>
Team Roles .....	5
Adaptation of the Agile Scrum Methodology .....	6
<b>5. Processing Workflow and Technical Architecture.....</b>	<b>8</b>
General system workflow .....	8
Used Technologies.....	9
System infrastructure and architecture .....	9
<b>6. System Functionalities .....</b>	<b>10</b>
General Functionalities .....	10
Specific functionalities by user profiles .....	10
<b>7. Results and Demonstration.....</b>	<b>12</b>
Landing page .....	12
Examples of chatbot requests and responses .....	12
Knowledge graph visualization .....	14
Use cases by user profiles.....	15
Qualitative evaluation of performance and utility .....	18
<b>8. Impact and Relevance.....</b>	<b>19</b>
Scientific impact.....	19
Technological impact.....	19
Social and global impact .....	20
Potential impact in future missions .....	20
<b>9. Future and next steps.....</b>	<b>20</b>
<b>Appendix A. GraphRAG: Graph-Based Retrieval-Augmented Generation (Deep Dive).....</b>	<b>22</b>
<b>Appendix B. Implementation details of the Assay Finder (Researcher functionality) .....</b>	<b>26</b>
<b>Appendix C. Implementation details of the Gap Finder (Investor functionality).....</b>	<b>29</b>
<b>Appendix D. Implementation details of the Mission Planner (Mission Architect functionality) .....</b>	<b>33</b>





## 1. Executive Summary

The **Stellar Mind AI project**, developed by the **Stellar Minds team** to address the “**Build a Space Biology Knowledge Engine**” challenge, proposes a new way to explore the biological knowledge generated by **NASA** over decades of space research. Thousands of experiments on how humans, plants, and microorganisms respond to the space environment have been published in scientific papers, but their volume and diversity make it difficult to access and fully understand this information. **Our goal is to transform this scattered dataset into a dynamic, interactive, and accessible knowledge base powered by artificial intelligence and knowledge graphs.**

To achieve this, we use **GraphRAG**, an architecture that combines retrieval-augmented generation with semantic representation in graphs. From the **608 bioscience documents provided by NASA**, we processed the texts, extracted key concepts, relationships, and relevant biological entities, and stored them in **Parquet format** for efficient querying. **The result is a navigable knowledge graph and an intelligent chatbot** that allows users to query scientific findings in natural language.

**The web application integrates two general functionalities** (graph navigation and conversational chat with the knowledge base) **and three specific tools** designed according to the user profiles identified by NASA:

- **Researchers:** *Assay Finder*, a tool that facilitates the search for experiments, results, and methodologies without requiring technical knowledge of APIs.
- **Investors and managers:** *Gap Finder*, which detects trends, investment opportunities, and gaps in space biology research.
- **Mission architects:** *Mission Planner*, which analyzes mission objectives and provides recommendations supported by scientific facts and well-founded arguments.

With this platform, **we aim to accelerate scientific discovery, improve space mission planning, and democratize access to NASA-generated knowledge**, aligning with the principles of open science and sustainable space exploration. **In this way, we combine scientific rigor with technological innovation, contributing to a new era of interdisciplinary research driven by artificial intelligence.**

Finally, it is worth emphasizing that **this project is not a showcase of empty promises nor a display of synthetic data**, but a system that the reader can test right now, with **real data provided by NASA**. It should be noted that **addressing this challenge using cutting-edge architectures such as GraphRAG represents a significant technical achievement**, and every element described in this report functions exactly as stated. **All of this functionality can be tested at the following link:**

<https://vercel-app-frontend-tawny.vercel.app/>



## 2. Context and Motivation

The “Build a Space Biology Knowledge Engine” challenge proposed by NASA seeks to enable a new era of human space exploration through the intelligent use of scientific data generated over decades of biological research in space environments. From the first microgravity experiments to the most recent studies on the **International Space Station**, NASA has accumulated a vast collection of results on how living organisms (**humans, plants, and microorganisms**) respond to the extreme conditions of space. **This information is essential to prepare future sustainable missions to the Moon and Mars**, but it is currently scattered across hundreds of scientific publications that are difficult to integrate and analyze in a unified way.

The Stellar Minds team recognizes that one of the greatest challenges for the scientific and technological community is **efficient access to space biology knowledge**. Although the data are publicly available, fragmentation across databases, the growing volume of publications, and the lack of interactive analytical tools make it difficult for researchers, managers, and engineers to extract relevant conclusions or identify gaps in research. **This gap between available information and its practical use limits the ability to generate new hypotheses, optimize research investments, or anticipate risks in future missions.**

In this context, **Stellar Mind AI** emerges as a response to that need: **an interactive and dynamic knowledge base powered by artificial intelligence and knowledge graphs** that enables the organization, synthesis, and intuitive exploration of NASA’s vast corpus of space biology publications. **The platform leverages emerging technologies such as GraphRAG** to connect data, concepts, and discoveries, thereby facilitating a deeper understanding of biological interrelations in the space environment.

Through this initiative, **the Stellar Minds group seeks not only to improve access to scientific knowledge but also to promote informed decision-making in key areas of space exploration**—from planning safe missions to identifying strategic research and investment opportunities. **Ultimately, the Stellar Mind AI project aims to transform the way space biological knowledge is discovered, interpreted, and applied, contributing to the global objective of preparing humanity to live and thrive beyond Earth.**

## 3. Project Goals

### General objective

The **Stellar Mind AI** project, developed by the Stellar Minds team, has the overall objective of creating an intelligent platform that **synthesizes, organizes, and makes accessible** the **space biological knowledge** generated by NASA through the use of **artificial intelligence, knowledge graphs**, and advanced **natural language processing (NLP)** techniques. This tool aims to enhance scientific and technological **space exploration**, enabling different user profiles — **researchers, investors, and mission architects** — to query, analyze, and discover relevant information in an intuitive and meaningful way.



## Specific objectives

- **Build an interactive knowledge base using GraphRAG.** Based on 608 NASA bioscience documents, process, extract, and structure entities, relationships, and key concepts into a semantic graph stored in Parquet format for efficient and scalable querying.
- **Develop an intuitive web interface for graph querying and navigation.** Implement an application that allows users to visually explore the connections between experiments, results, organisms, and space conditions, as well as interact through a conversational chatbot capable of answering natural language questions based on the knowledge graph.
- **Design intelligent search functionalities tailored to three key user profiles:**
  - o **Researcher – Assay Finder:** a tool to access experimental information, results, and methodologies without requiring technical knowledge of APIs or databases.
  - o **Investor – Gap Finder:** modules that identify emerging areas, research trends, and scientific gaps representing strategic investment opportunities.
  - o **Mission Architect – Mission Planner:** a module that uses NASA's bioscience data and AI reasoning to create evidence-based mission plans from simple natural language queries.

Through these objectives, Stellar Mind AI seeks to integrate the power of **artificial intelligence** with **NASA's open science**, driving discovery, interdisciplinary collaboration, and knowledge-based decision-making in the field of space biology.

## 4. Work Methodology

### Team Roles

- **Candela Dávila Moreno – Head of Design:** Responsible for the visual design and aesthetic identity of the project. Her work ensures a coherent, engaging visual experience aligned with the scientific objectives of the product.
- **Eulogio Quemada Torres – CTO:** In charge of the technical architecture and infrastructure decisions. He oversees the integration of the system's different modules and ensures backend efficiency and scalability.
- **Javier Montes Pérez – Head of AI:** Leads the development of artificial intelligence components, including the implementation of GraphRAG, data processing, and integration of language models.
- **Alejandro Román Sánchez – Head of UX:** Designs the user experience, ensuring intuitive navigation and smooth interaction with the chatbot and knowledge graph visualizations.
- **Macorís Decena Giménez – CPO and Scrum Master:** Coordinates the agile planning of the project, task prioritization, and alignment among design, development, and artificial intelligence. Oversees the product vision and value delivery.

In addition, all team members have worked as Full Stack developers, adapting to the project's needs at different stages of development.





## Adaptation of the Agile Scrum Methodology

At **Stellar Minds**, we used an **agile methodology based on Scrum**, adapted to the intensive context of the hackathon. We applied an **Express version of the framework**, organizing the work into **one-day sprints**, which allowed us to maintain a fast development pace and constant communication among team members. Each day began with a **brief planning meeting** and ended with a **review of progress and blockers**. **Tasks were managed through a Kanban board** structured in three columns: **“Backlog,” “In Progress,”** and **“Done.”** This visual approach facilitated organization, task prioritization, and simultaneous collaboration among **design, development, and artificial intelligence**.

Thanks to this **agile methodology**, the team was able to **iterate quickly, maintain a shared vision of the project**, and **ensure continuous value delivery** throughout all phases of **Stellar Mind AI’s development**.

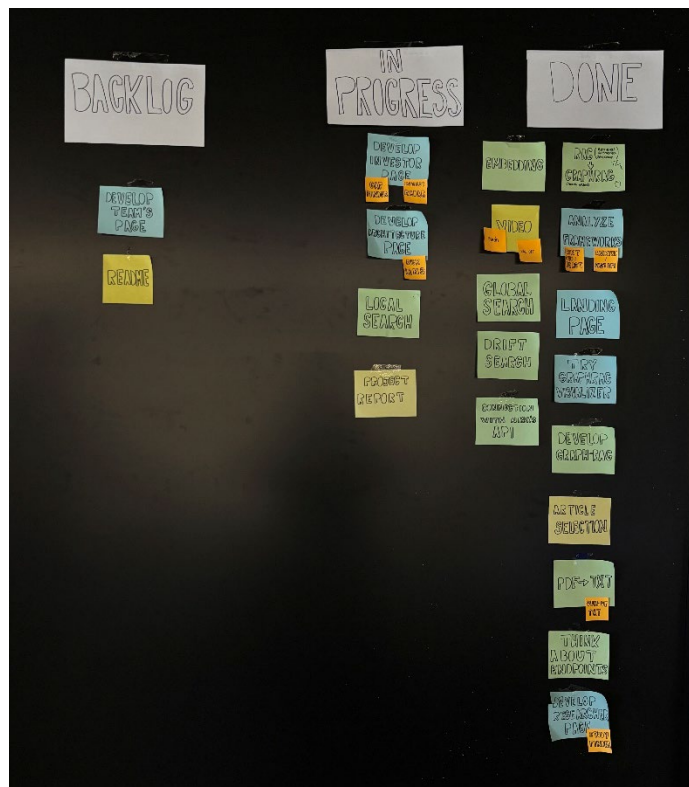
Below is a **sequence of three images**, each representing the **progressive evolution** of the **physical Kanban board** that we decided to build in our workspace. Since the team had three working days, we structured the schedule into **three one-day sprints**.



*Backlog halfway through the first day of work (Sprint 1/3).*



Backlog halfway through the second day of work (Sprint 2/3).



Backlog halfway through the third day of work (Sprint 3/3).





## 5. Processing Workflow and Technical Architecture

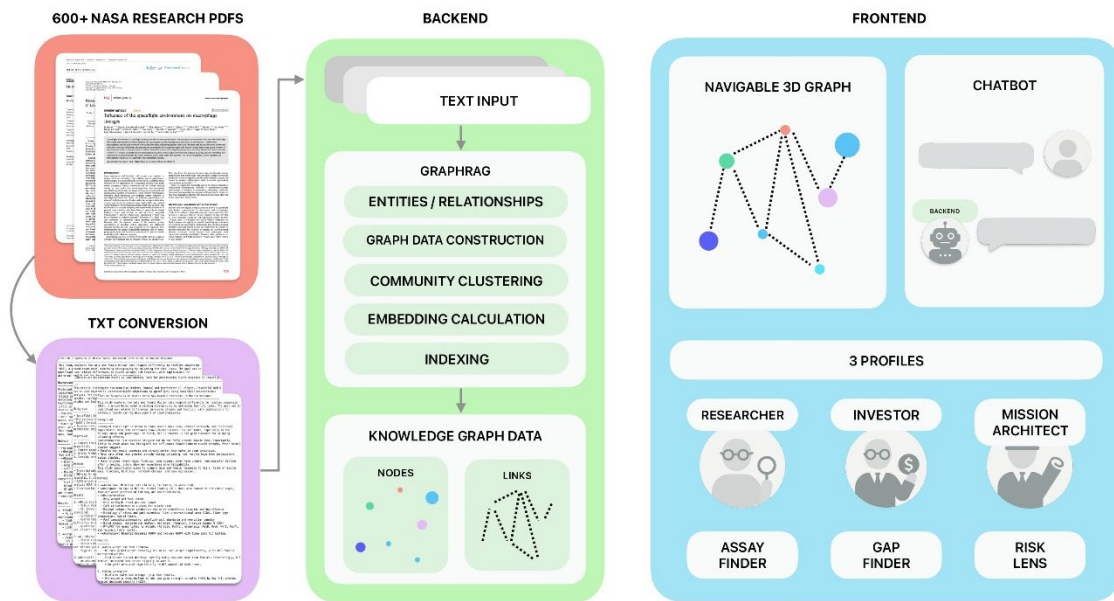
The **Stellar Mind AI project** is built on a **modular architecture** designed to **process large volumes of scientific information**, transform them into **structured knowledge**, and make them accessible to users through an **interactive and user-friendly web interface**. The methodology combines **natural language processing (NLP)**, **knowledge graphs**, and **large language models (LLMs)** to enable **semantic queries and advanced inferences** over NASA's **space biology literature**.

### General system workflow

The system pipeline is composed of five main stages, integrated in a sequential and automated manner:

1. **PDF Processing:** The space biology documents provided by NASA are preprocessed using text extraction and data cleaning tools. Redundant elements (tables, references, metadata) are removed, and the content is segmented into relevant sections (Introduction, Results, Conclusions) to maximize semantic quality.
2. **Knowledge Extraction with GraphRAG:** GraphRAG (Graph-based Retrieval-Augmented Generation), an open-source technology by Microsoft, is used to analyze the texts and extract entities, relationships, and key concepts. This stage generates nodes and edges that represent explicit and implicit knowledge within the documents, connecting experiments, organisms, biological variables, and findings.
3. **Parquet File Generation:** All structured information is stored in Apache Parquet format, optimized for fast queries and efficient storage. This enables the handling of large volumes of scientific data in a scalable and reproducible way.
4. **Semantic Graph Construction:** The processed data are integrated into a knowledge graph that facilitates both visual and semantic exploration of the relationships among studies, results, and research areas. This graph constitutes the core of the system and powers both the chatbot and the interactive visualizations.
5. **Web Interface (Chatbot + Interactive Visualization):** An intuitive web interface is developed that allows users to:
  - Query the graph using natural language through an LLM-based chatbot.
  - Visually explore nodes and connections in an interactive environment, identifying thematic areas, experimental relationships, and research trends.

The following diagram shows the key details of the processing workflow performed by Stellar Mind AI.



*Stellar Mind AI Workflow Diagram*

## Used Technologies

- **GraphRAG**: knowledge extraction tool and semantic graph generation framework.
- **Language Models (LLMs)**: used to summarize, classify, and answer complex questions.
- **Python**: main language for the backend and processing pipeline.
- **FastAPI**: backend framework that manages chatbot queries, the search API, and communication with the graph; known for its performance, modern typing, and scalability.
- **React**: frontend framework used to build a dynamic, modular, and highly interactive interface, ideal for graph visualization and real-time communication with the API.
- **Vercel**: frontend deployment platform that enables continuous integration, fast loading speeds, and global reliability.
- **Apache Parquet and SQLite**: used for structured storage and optimized local queries.

## System infrastructure and architecture

The system is organized under a **distributed client-server architecture**, where the **backend** (FastAPI) serves as the access point to the **data processed by GraphRAG**, and the **frontend** (React) consumes the APIs to display the information.

- The **backend** endpoints allow querying the graph, retrieving relevant entities or relationships, and feeding the conversational engine.
- The **frontend**, deployed on Vercel, communicates user requests and renders the results visually and interactively.
- The **knowledge graph** and generated embeddings are hosted in a Python processing environment, ensuring independence and modularity.



## Scalability and Reproducibility

The system was designed with **horizontal scalability principles**, allowing the volume of processed publications to be expanded without compromising performance. The use of **open formats** (Parquet, JSON, GraphML) and **reproducible technologies** (FastAPI, React, GraphRAG) ensures **project portability** and **scientific replicability**. Moreover, the **modular architecture** facilitates future integration with other **NASA data sources**, such as the **Open Science Data Repository (OSDR)** or the **Space Life Sciences Library**.

## 6. System Functionalities

The **Stellar Mind AI project** is designed to provide an **accessible, visual, and intelligent scientific exploration experience**. Its **modular architecture** allows different user profiles — from **researchers to mission architects** — to interact with the same **knowledge base** from different perspectives, adapting functionalities to their **specific needs**.

### General Functionalities

These tools form the **core of the application** and are **shared by all system users**.

- **Natural Language Query Chatbot:** A virtual assistant powered by **large language models (LLMs)** allows users to make natural language queries about the processed space biology knowledge. The chatbot can **answer questions, summarize studies, identify connections between investigations, and suggest relevant sources**. Thanks to its integration with the **semantic graph**, responses go beyond simple text matching — they reflect **conceptual and contextual relationships**, providing a **deeper and more scientific understanding** of the data.
- **Interactive Knowledge Graph Visualization:** The web application includes a **visualization module** that represents the **concepts, entities, and relationships** extracted from NASA's 608 biological documents.

Through an **interactive interface developed in React**, users can **navigate the graph, filter information** by topic or entity type, and **discover emerging connections** among experiments, results, and organisms.

This functionality promotes **intuitive knowledge exploration** and allows users to **observe how different studies interrelate within a global context**.

### Specific functionalities by user profiles

To maximize the **impact and applicability** of the system, **Stellar Mind AI** offers **three sets of tools** tailored to key profiles within the **scientific and technological ecosystem of space exploration**.



### **a. Researcher – “Assay Finder”**

The **Assay Finder** module facilitates **easy and fast access** to experimental information. It allows researchers to **search for experiments, methodologies, results, or mission conditions** without needing to understand NASA’s internal API structure. The system interprets **complex queries**, locates **relevant experiments**, and provides **structured summaries** that include the type of assay, key results, microgravity conditions, studied organisms, and main conclusions.

In this way, it **accelerates scientific literature review** and promotes the **formulation of new research hypotheses**.

### **b. Investor – “Gap Finder”**

The **Gap Finder** module analyzes relationships within the graph to detect **knowledge gaps** — underexplored topics, unstudied combinations of biological variables, or missing experimental results. This tool offers **proactive recommendations for future research lines**, promoting **scientific innovation and interdisciplinary collaboration**.

### **c. Mission Architect – “Mission Planner”**

The **Mission Planner** module is designed to assist in the **conceptualization and detailed planning of space missions** using NASA’s accumulated bioscience knowledge. Through natural language queries, users can describe a mission scenario — for example, *“Plan a 30-day Mars analog to test immune countermeasures”* — and the system automatically generates a structured mission brief. This includes **scientific objectives, risk posture, recommended countermeasures**, and **supporting evidence** derived from NASA’s research data.

By leveraging **graph-based reasoning** through GraphRAG, Stellar Mind AI connects previous biological experiments, environmental studies, and spaceflight data to inform each plan. The result is a data-driven tool that helps mission architects **anticipate challenges, optimize resource allocation**, and **design safer, evidence-based operations** for lunar and Martian exploration.

Together, these functionalities make **Stellar Mind AI** a versatile, interactive, and scientifically grounded platform that merges artificial intelligence with space bioscience — fully aligned with NASA’s commitment to **open, collaborative, and forward-looking research**.



## 7. Results and Demonstration

The **Stellar Mind AI project**, developed by the **Stellar Minds team**, culminates in a **fully functional web application** that combines a **knowledge base processed through GraphRAG**, an **intelligent chatbot**, and an **interactive visualization of the knowledge graph**.

The results demonstrate the system's ability to **integrate complex information**, **generate useful responses**, and provide an **intuitive visual exploration of space biology data**.

### Landing page

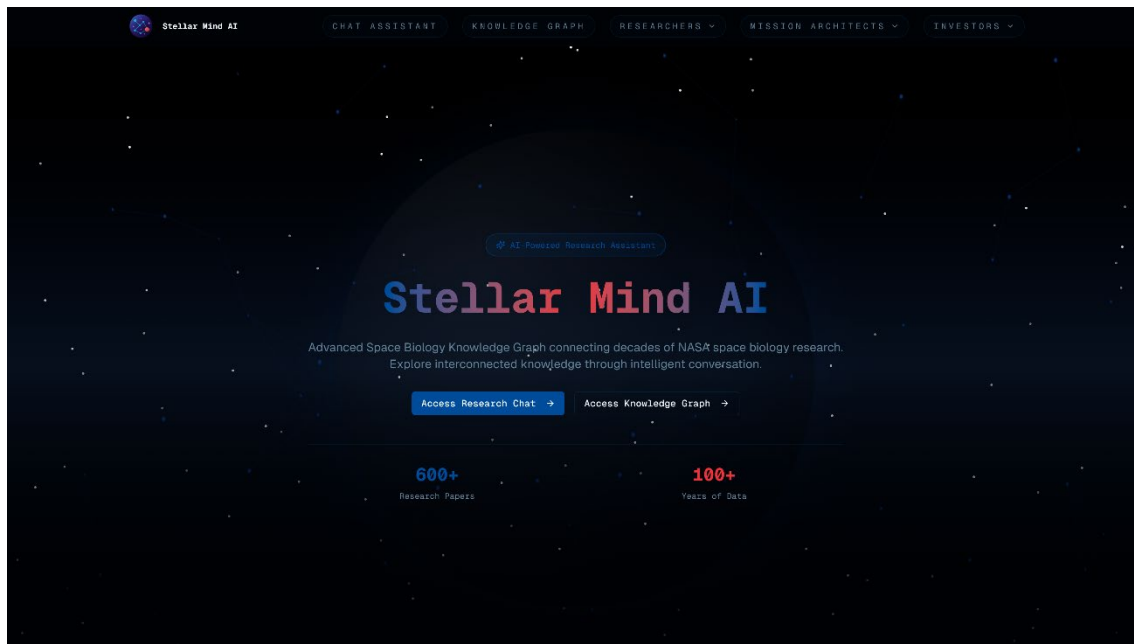
The **public URL** to access this functionality is:

<https://vercel-app-frontend-tawny.vercel.app/>

The system's **main interface** features a **dynamic dashboard**, where the user can switch between the following modes:

- **Conversational Chat:** interaction in **natural language** with the knowledge base.
- **Graph Visualization:** exploration of **nodes, relationships, and information clusters**.

Screenshot of the landing page:



### Examples of chatbot requests and responses

The **public URL** to access the general chatbot's functionality is:

<https://vercel-app-frontend-tawny.vercel.app/chat>



The **Stellar Mind AI chatbot** demonstrates the ability to perform **graph-based inferences**, providing **well-founded answers** and **references to the original studies**.

Some representative examples include:

Tell me about space research with the human species

Space research involving the human species encompasses a wide range of studies aimed at understanding the effects of spaceflight conditions on human health and physiology. These studies often focus on aspects such as the astronaut microbiome, bone health, ocular health, immune function, and overall well-being during space missions.

One significant area of research involves the Astronaut Microbiome, which refers to the complex ecosystem of bacteria, fungi, viruses, and archaea within the human body. This microbiome plays a crucial role in immunity, metabolism, cognition, and overall physiological balance. Conditions experienced during space missions, such as microgravity, radiation, psychological stress, and altered diets, can impact the composition and function of the microbiome, potentially leading to health issues like inflammatory bowel disease, obesity, asthma, and diabetes [Data: Reports (20), Entities (399), Relationships (149, 150, 151, 163, 176, 181)].

Another critical area of study is bone health in spaceflight. Research has shown

countermeasures to safeguard astronaut health and well-being during long-duration space missions.

Now tell me about gravity-related studies with mice

Studies related to gravity and its effects on mice have been conducted in various contexts, including spaceflight, simulated microgravity, and artificial gravity. One study titled "Mouse Brain Study on Gene Expression Changes" focused on investigating the gene expression alterations in the mouse brain under spaceflight conditions and the impact of antioxidant treatment with BuOE [Data: Reports (51)]. The study highlighted the significance of the Illumina NovaSeq sequencing platform in analyzing the effects of spaceflight exposure on gene expression in the mouse brain [Data: Entities (590)]. It also emphasized the role of different groups of mice, such as ground control and flight groups, in understanding the effects of spaceflight and antioxidant treatment on gene expression [Data: Entities (586, 587, 588, 589)]. The findings suggested that spaceflight induces gene expression changes in the mouse brain and that

These responses go beyond simple text extraction — they **combine information from multiple documents** and **reason over scientific relationships** present within the graph.





## Knowledge graph visualization

The **public URL** to access the knowledge graph functionality is:

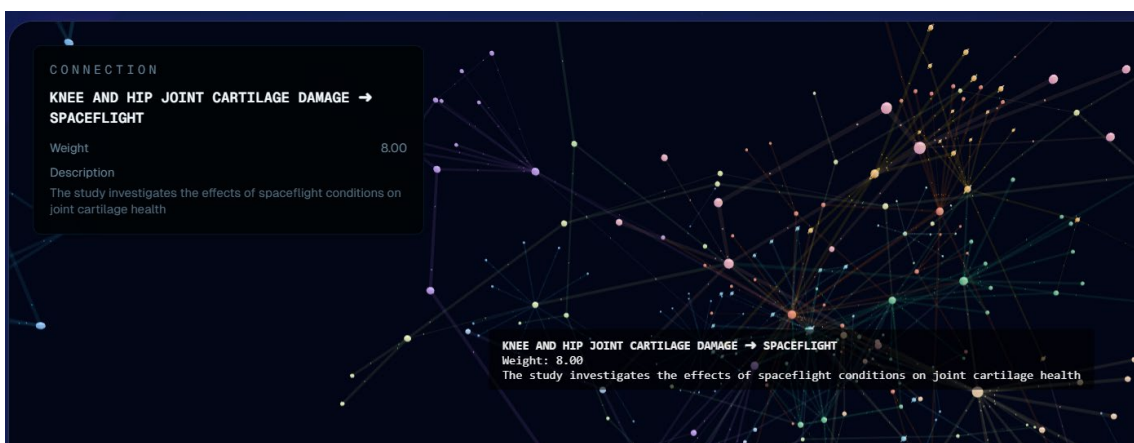
<https://vercel-app-frontend-tawny.vercel.app/graph>

The **Stellar Mind AI Knowledge Graph** enables users to visually and interactively navigate the information extracted from all NASA bioscience documents processed through GraphRAG. Each **node** represents a semantically identified entity — such as an *organism*, *experimental condition*, or *biological process* — while each **edge** represents a relationship between them, for example, “*microgravity affects bone density*” or “*radiation exposure alters gene expression*.” This dynamic graph allows users to explore how studies, experiments, and discoveries connect across decades of research, transforming complex scientific data into an intuitive visual network of knowledge.

The following image shows a representative view of the knowledge graph:



*General view of the Knowledge Graph, with information about more than 600 research articles from NASA.*



*View of a selected node, showcasing its information.*



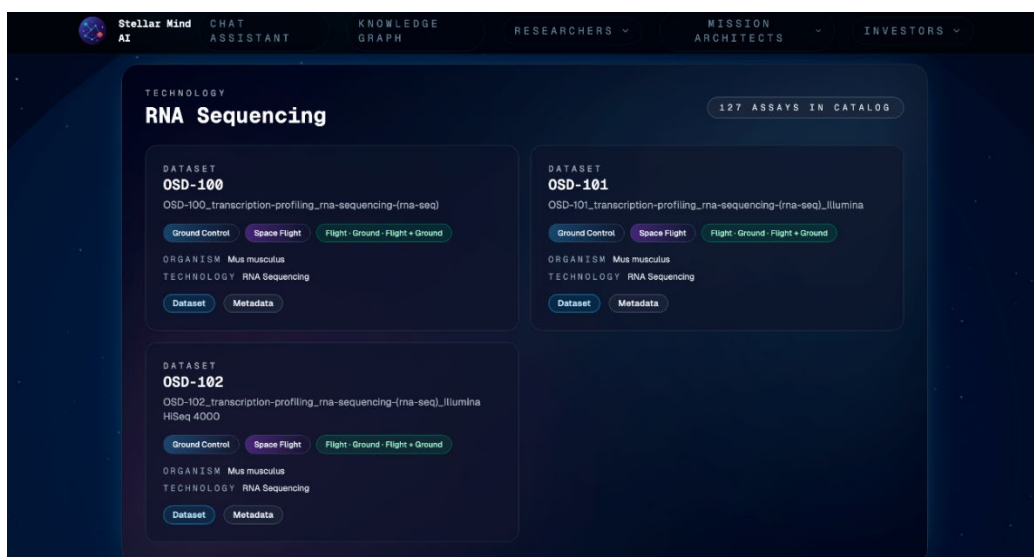
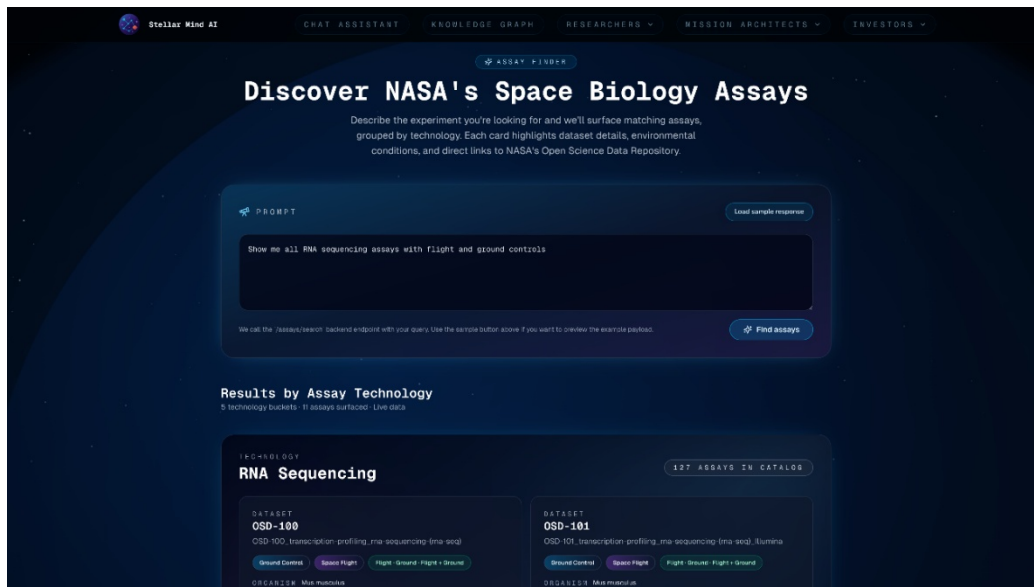
## Use cases by user profiles

### a. Researcher – Assay Finder

The **public URL** to access this functionality is:

<https://vercel-app-frontend-tawny.vercel.app/assay-finder>

In the following screenshot demonstration, the **Assay Finder** module is used to query NASA's bioscience database with the prompt *"Show me all RNA sequencing assays with flight and ground controls."* The system processes the request and returns a structured list of relevant experiments, grouped by assay technology. Each result card displays key metadata such as the **dataset ID**, **organism studied** (*Mus musculus*), and **experimental conditions** (Ground Control, Space Flight, Flight + Ground). This visualization illustrates how Stellar Mind AI automatically interprets a complex research query and retrieves precise, context-rich scientific information, helping researchers quickly identify comparable experiments and supporting the design of new studies based on existing NASA data.

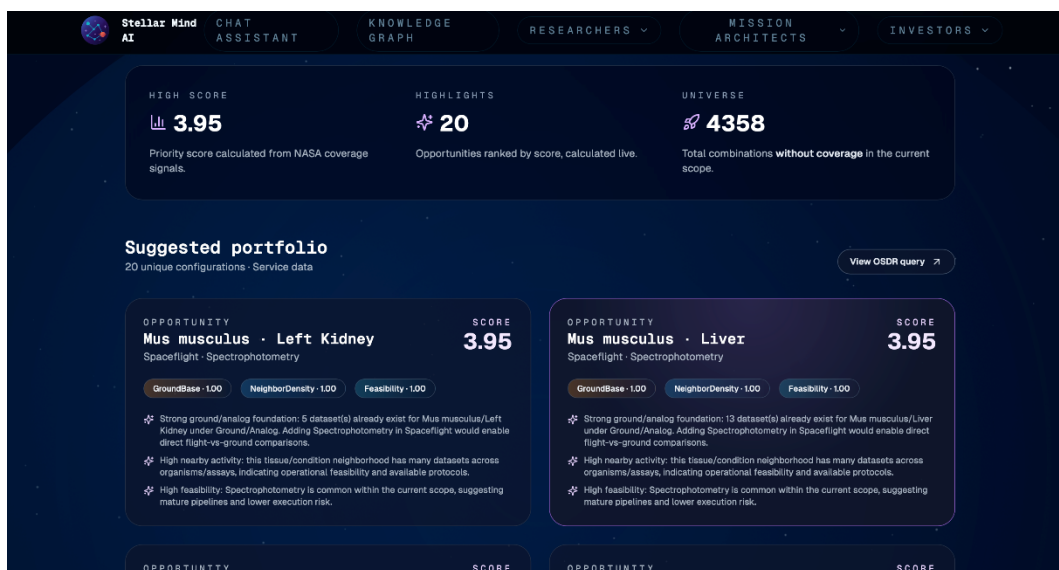
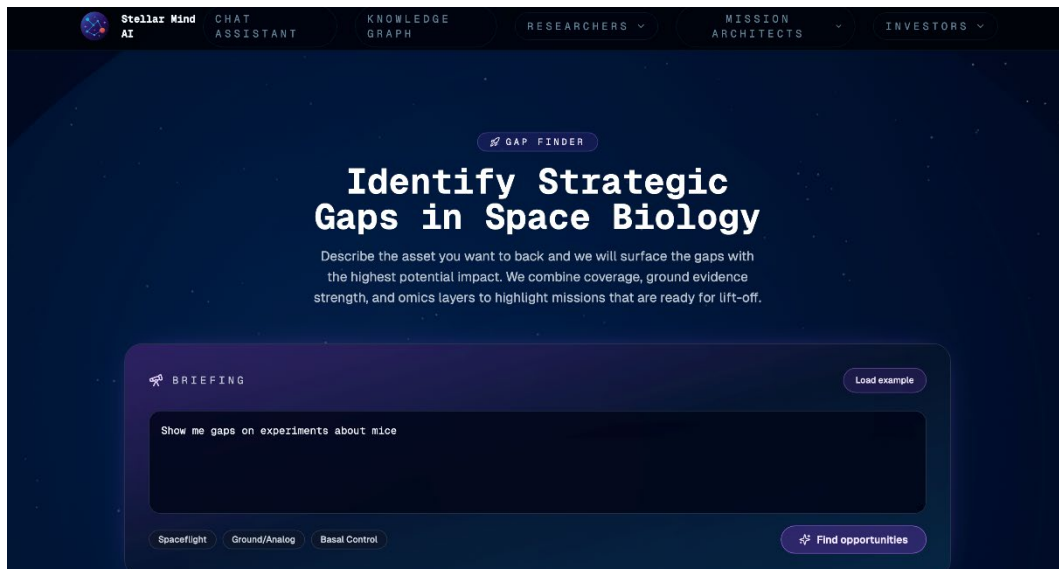


**b. Investor – Gap Finder**

The **public URL** to access this functionality is:

<https://vercel-app-frontend-tawny.vercel.app/gap-finder>

In this demonstration, the **Gap Finder** module is used to identify underexplored areas in NASA’s bioscience research. The user submits the query “*Show me gaps on experiments about mice*”, and the system analyzes the coverage density of existing datasets within the GraphRAG knowledge base. The interface displays **key indicators** such as the priority score (3.95), the number of highlighted opportunities (20), and the total combinations without coverage (4,358). Below, the suggested portfolio lists specific research opportunities — for example, *Mus musculus – Left Kidney* and *Mus musculus – Liver* under *Spaceflight* conditions — along with feasibility metrics and evidence-based rationales. This visualization demonstrates how Stellar Mind AI supports investors and research strategists in detecting potential knowledge gaps, uncovering new experimental opportunities, and guiding strategic decisions for future investment in space biology research.



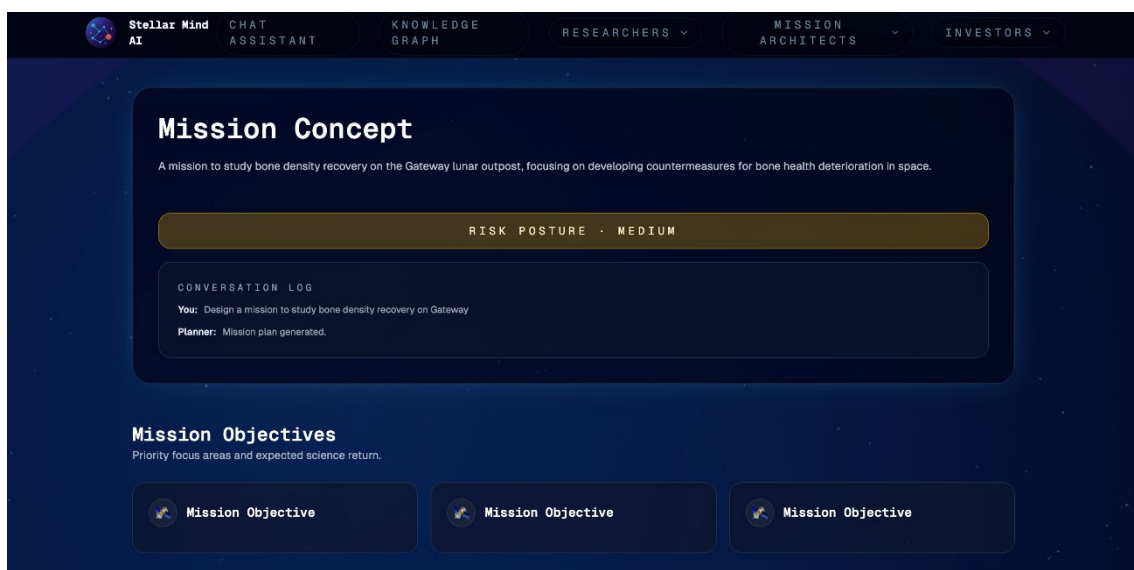
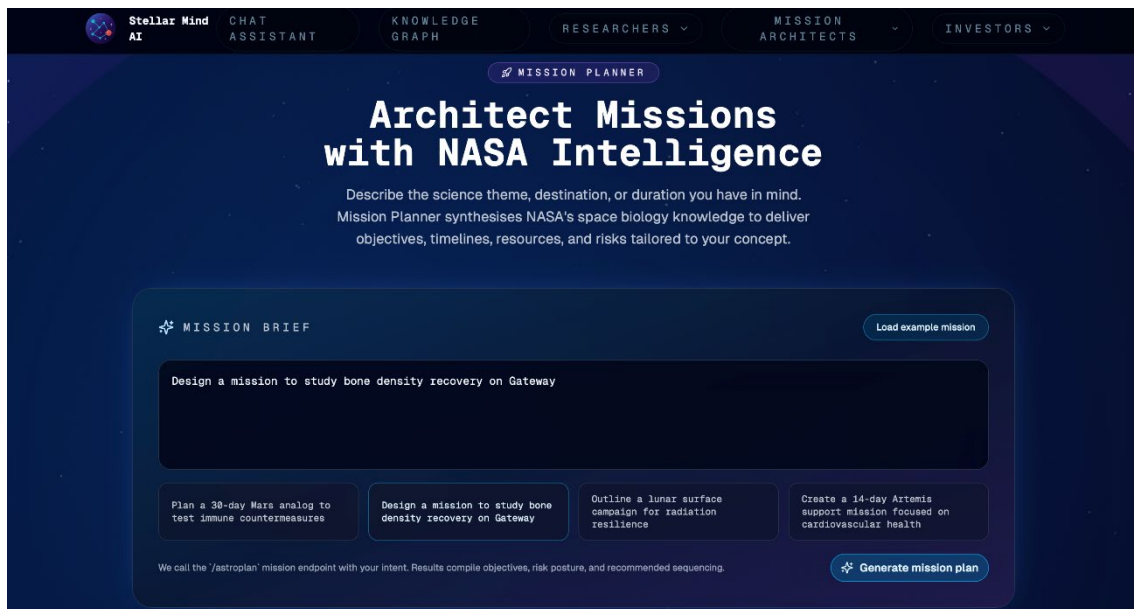


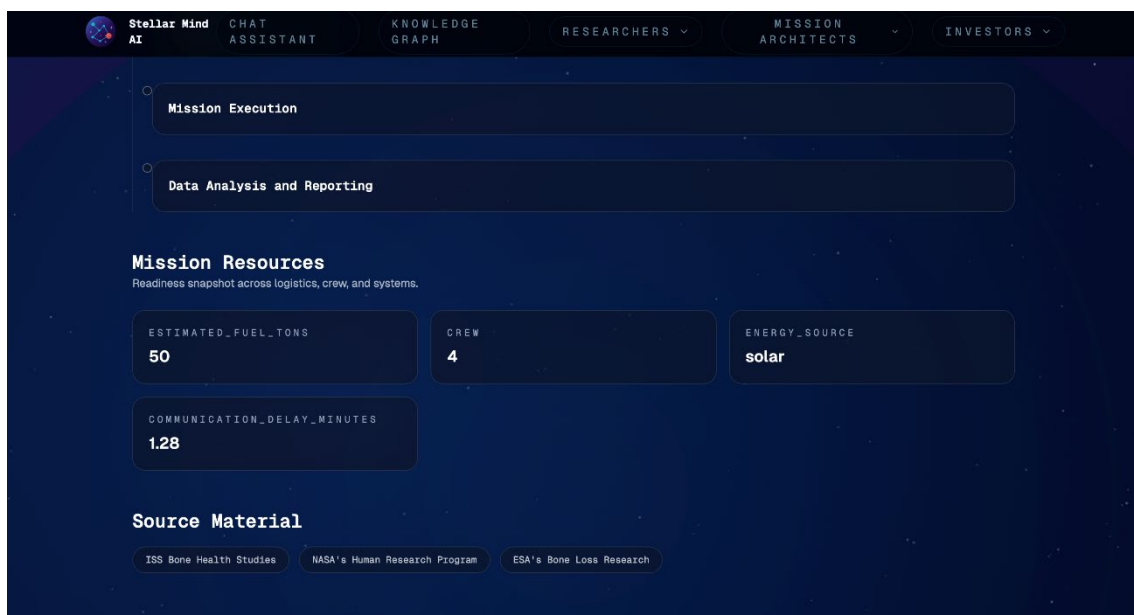
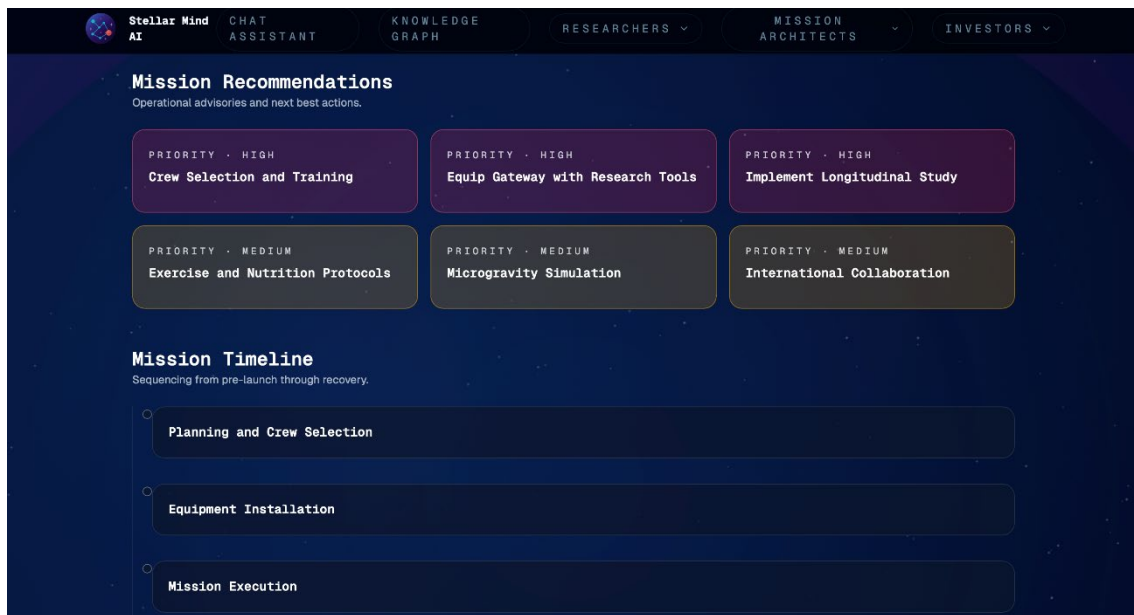
### c. Mission Architect – Mission Planner

The **public URL** to access this functionality is:

<https://vercel-app-frontend-tawny.vercel.app/mission-planner>

The **Mission Planner** module enables users to design and analyze **space mission concepts** using **NASA’s bioscience knowledge** through **natural language input**. From a single query—such as “*Design a mission to study bone density recovery on Gateway*”—the system generates a **structured mission plan** that includes objectives, risk posture, recommendations, and required resources. The outputs displayed in the interface summarize the **Mission Concept**, list **scientific objectives** and **operational priorities**, define a **timeline** from planning to data analysis, and estimate **key logistics** such as fuel, crew size, and communication delay. Each element is derived from **graph-based reasoning with GraphRAG**, linking experimental data and prior research to support **evidence-driven, safe, and optimized mission planning**.





## Qualitative evaluation of performance and utility

During internal testing, **Stellar Mind AI** demonstrated a **high degree of semantic coherence** and **contextual relevance** in the responses it generated. The system responds **fluently to complex queries** and is capable of **integrating results from multiple documents**, showcasing the effectiveness of the **GraphRAG** approach for representing scientific knowledge.

The **graph visualization** allows users to identify **dense research areas** and **knowledge gaps**, providing added value to both **scientists** and **innovation managers**.





From the user's perspective, the main observed strengths are:

- **Ease of use:** natural interaction without requiring technical knowledge.
- **Interpretability:** relationships and sources are displayed transparently.
- **Versatility:** the tool adapts to different user profiles and use cases.

The results confirm that **Stellar Mind AI** represents an **innovative, practical, and scalable solution** for transforming **space biological knowledge** into an **accessible, understandable, and useful information network** for the scientific and technological community.

## 8. Impact and Relevance

The **Stellar Mind AI project**, developed by the **Stellar Minds team**, represents a **significant contribution** at the intersection of **space biological science** and **artificial intelligence**. Its impact spans **scientific, technological, social, and operational dimensions**, all directed toward a common goal: **accelerating the discovery and understanding of the biological knowledge essential for sustainable human space exploration**.

### Scientific impact

**Stellar Mind AI** substantially improves the **access, understanding, and reuse** of space biological knowledge. Through the **automatic extraction of information** from more than **600 scientific publications** and its organization into a **semantic graph**, the system enables researchers and analysts to quickly find **relevant results, correlations, and gaps** within the literature.

This capability transforms the traditional approach to **literature review and comparative analysis**, providing a **more efficient, integrative, and evidence-based** environment for scientific exploration.

Furthermore, by structuring knowledge in an **open and visual manner**, **Stellar Mind AI** facilitates the identification of **experimental trends, emerging hypotheses, and interdisciplinary relationships**, thereby strengthening **data-driven science** applied to **space biology**.

### Technological impact

From a **technological perspective**, the project stands out for its **innovative use of artificial intelligence** applied to **complex scientific data**.

By integrating **GraphRAG** with a **FastAPI processing pipeline** and an **interactive React interface**, **Stellar Mind AI** combines **cutting-edge technologies** in **knowledge representation, semantic search, and graph visualization**.

This approach not only demonstrates the **feasibility of using LLMs and knowledge graphs** in real scientific environments but also **lays the groundwork for future platforms** dedicated to **automated scientific literature analysis**.





The system's **modular, reproducible, and scalable architecture** ensures its applicability to other scientific domains, establishing **Stellar Mind AI** as a model of **technological innovation in the service of open knowledge**.

### Social and global impact

The project directly contributes to the **democratization of space knowledge** by providing the **scientific, educational, and technological community** with an **open, interactive, and easy-to-use tool**. Through the use of **natural language interaction** and **intuitive visualizations**, **Stellar Mind AI** breaks technical barriers, allowing users without specialized training in data mining or programming to access the accumulated knowledge from **decades of space research**.

In doing so, the initiative advances the **values of open science, interdisciplinary collaboration, and equitable access to knowledge**, fostering the **spirit of global cooperation** that defines the **NASA Space Apps Challenge** program.

### Potential impact in future missions

The potential of **Stellar Mind AI** extends **beyond scientific analysis**, offering a powerful tool to support the **planning, design, and evaluation of space missions**.

Through the **Mission Planner** module, the system can automatically generate structured **mission plans** from **natural language input**, outlining **objectives, risks, countermeasures**, and supporting **evidence** drawn from NASA's bioscience data.

Similarly, the **Gap Finder** functionality can assist **agencies, universities, and innovation centers** in **shaping their research strategies** toward areas with **greater potential impact** in **lunar and Martian exploration**.

Together, these components make **Stellar Mind AI** a **strategic tool** that bridges **biological science** and **mission engineering**, strengthening the ecosystem for **future space exploration**.

In conclusion, **Stellar Mind AI** does not merely organize data — it **transforms dispersed information into actionable knowledge**, fostering **scientific research, technological innovation**, and the **safe planning of human exploration beyond Earth**.

## 9. Future and next steps

The current version of **Stellar Mind AI** already provides a solid and fully functional foundation, with all core components — including the knowledge graph, intelligent chatbot, and dedicated modules for each user profile — available on the live web platform. Building on this robust base, future iterations could incorporate **additional specialized functionalities** tailored to each user group, deepening the system's analytical power and broadening its scientific and operational scope. These new tools would enhance the decision-making capabilities of researchers, investors, and mission architects alike, further aligning Stellar Mind AI with NASA's vision of **open, data-driven, and intelligent exploration**.



- **Researchers – “Hypothesis Builder”**

This tool assists scientists in **formulating and validating new research hypotheses** based on NASA’s bioscience corpus. By analyzing co-occurrences, correlations, and experimental gaps in the knowledge graph, it suggests **potential causal relationships** (e.g., “radiation → gene expression changes → immune suppression”) and provides supporting evidence from prior studies. Researchers can iteratively refine hypotheses and export them as structured summaries for experimental planning or proposal drafting.

- **Investors – “Impact Forecaster”**

Designed to complement Gap Finder, this module uses **trend analysis and predictive modeling** to estimate the **future scientific or commercial impact** of specific research domains. By combining publication growth rates, collaboration networks, and citation patterns, it projects **emerging investment opportunities** (e.g., regenerative biology in microgravity) and **strategic timeframes** for funding. It provides dashboards with metrics such as “expected maturity level,” “impact score,” and “innovation velocity.”

- **Mission Architects – “BioSystems Simulator”**

This interactive simulation environment allows mission architects to **model biological systems under spaceflight conditions**, using empirical data from the knowledge graph. Users can input mission parameters (duration, radiation exposure, habitat type, etc.) and visualize predicted **biological responses** based on prior experiments. The simulator outputs dynamic models and “what-if” scenarios, helping to test hypotheses, design countermeasures, and assess mission safety before actual deployment.



## Appendix A. GraphRAG: Graph-Based Retrieval-Augmented Generation (Deep Dive)

### Background: Retrieval-Augmented Generation (RAG) and Its Limits

Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by retrieving external information relevant to a user's query and including it in the prompt. In a typical RAG system, when a user asks a question, the system searches a knowledge source for relevant information and augments the LLM's input with those results. This helps ground the LLM's answers in factual content, mitigating hallucinations and extending knowledge beyond the model's training data.

Most RAG systems use vector similarity search to find relevant documents or text chunks. Each document is split into chunks, embedded as high-dimensional vectors, and stored in a vector index. At query time, the query is also embedded, and the closest matching vectors are retrieved.

However, baseline RAG has limitations:

- **Fragmented context:** It finds similar passages but doesn't capture relationships between them, leading to incomplete answers.
- **Multi-hop reasoning challenges:** It struggles to answer complex questions that require connecting information across documents.
- **Lack of explainability:** It's hard to explain why a specific chunk was retrieved, which reduces transparency and trust.

These challenges motivated integrating knowledge graphs into RAG, creating **GraphRAG**.

### What is GraphRAG?

GraphRAG integrates knowledge graph structures into the RAG workflow to improve retrieval and grounding of LLM responses. A graph of nodes (entities) and edges (relationships) serves as structured memory instead of relying solely on text similarity.

#### Key advantages:

- **Richer retrieval:** GraphRAG retrieves connected subgraphs of related information, uncovering multi-hop relationships.
- **Contextual augmentation:** The graph provides an explainable structure, allowing users to trace connections between facts and queries.

GraphRAG overcomes the weaknesses of vector-only RAG by considering relationships and global context, leading to more accurate, complete, and trustworthy answers.

### Internal Workflow and Components

GraphRAG has two main phases: **indexing (graph building)** and **query (retrieval and generation)**.



## 1. Indexing Phase: Building the Knowledge Graph

1. **Document ingestion and chunking:** Source documents are split into smaller text units.
2. **Entity and relationship extraction:** LLMs identify entities and relationships from text, producing triples like (Apollo, carries\_experiment, plant growth).
3. **Graph construction:** Entities become nodes, relationships become edges, and identical entities are merged.
4. **Graph storage:** The knowledge graph is stored in a graph database. Graph and vector indices can coexist.
5. **Community detection:** Clusters of related entities are identified using algorithms like Leiden to find thematic communities.
6. **Community summary generation:** LLMs summarize each cluster, creating thematic reports for later use.

At the end, GraphRAG has a graph of entities, relationships, clusters, and summaries ready for query time.

## 2. Query Phase: Graph-Based Retrieval and Augmented Generation

When a user asks a question:

1. **Query processing:** Identify entities in the query and classify the query as focused (local) or broad (global).
2. **Retriever – Graph-enhanced search:**
  - Find starting nodes using vector similarity.
  - Traverse the graph to gather connected information.
  - Filter and rank results based on textual similarity, graph distance, and relevance.
3. **Organizer (context assembly):** Organize retrieved facts and summaries coherently, preserving source references for explainability.
4. **Generation (LLM answering):** The LLM generates an answer using the organized context, often including citations.
5. **Feedback loop (optional):** The system can re-query if context is insufficient.

This synergy between graph search and LLM generation allows multi-hop, explainable reasoning.

## Mathematical and Algorithmic Details

- **Vector similarity retrieval:** Retrieve top-k similar embeddings for a query.



- **Knowledge graph representation:** Graph represented as nodes, edges, and adjacency matrix.
- **Graph traversal:** Perform depth-limited searches (1–2 hops) to collect related nodes.
- **Relevance scoring:** Combine semantic similarity and graph-based relevance to rank retrieved items.
- **Graph machine learning:** GraphRAG can use graph neural networks (GNNs) or knowledge graph embeddings to improve retrieval.
- **Community-based reasoning:** For broad queries, GraphRAG uses precomputed summaries from graph clusters.
- **Prompt composition:** Carefully balance raw text and summaries within the LLM's context window to maximize informativeness.

### Advantages of GraphRAG over Traditional RAG

- **Multi-hop and complex reasoning:** Connects related facts across documents.
- **Holistic dataset understanding:** Handles large-scale questions like “main themes in 600 research papers.”
- **Improved relevance and completeness:** Fills information gaps missed by vector search.
- **Reduced hallucination:** Graph grounding improves factual accuracy.
- **Explainability and traceability:** Every statement can be traced back to the graph and original sources.
- **Better context organization:** Context is structured and interconnected.
- **Integration with domain knowledge:** Can merge existing ontologies or databases for enriched reasoning.

Empirical evaluations show GraphRAG produces more comprehensive, diverse, and accurate answers than standard RAG.

### Example Use Case: NASA Bioscience Publications Dashboard

Given 600+ NASA bioscience publications, GraphRAG builds a knowledge graph of experiments, missions, results, and researchers. Clusters might represent themes like:

- Plant growth in microgravity
- Bone loss and countermeasures
- Microbial behavior in space
- Radiation effects on DNA



A user can ask, “What knowledge gaps remain for plant experiments in space?” GraphRAG retrieves the relevant cluster, explores connected nodes, and generates a detailed, cited answer summarizing known findings and gaps.

## Conclusion

GraphRAG represents a major step forward in integrating LLMs with structured knowledge. By combining graph reasoning with semantic retrieval, it produces grounded, explainable, and complete answers.

Its process:

1. Convert unstructured text into a structured knowledge graph.
2. Use the graph to retrieve interconnected context.
3. Feed that context to an LLM for grounded, multi-hop reasoning.

GraphRAG enables AI systems to reason semantically and contextually across vast datasets — ideal for knowledge-rich fields like NASA’s space biology research.





## Appendix B. Implementation details of the Assay Finder (Researcher functionality)

### Purpose

Assay Finder removes the friction of exploring spatial biology data in OSDR when the API is complex and the terminology heterogeneous. Starting from a free-text natural language input, the tool translates the researcher's intent into precise filters and queries the OSDR Biological Data API to return relevant assays, grouped by technology and accompanied by metadata and navigable links to the repository. Thus, a user who “thinks” in terms of organism, condition, and technique obtains in seconds an actionable list of experimental opportunities.

In OSDR, assays are experimental units (e.g., RNA-Seq, proteomics, microarrays, imaging) associated with datasets and samples within the ISA model (investigation–study–assay). The API offers specific “grouped by assay” endpoints that condense metadata by assay and make it easier to reason about their analytical utility and coverage.

### What problem does it solve?

**Barriers to entry:** The OSDR API supports powerful filters (OR with |, regex /.../i, presence filters =field), but its syntax imposes a learning curve. Assay Finder abstracts that complexity and auto-generates valid queries.

**Metadata heterogeneity:** The same semantics appear under different keys (tissue vs. organ, “pre/in/post-flight” variants). The tool normalizes those values for comparability.

**Assay-centric exploration:** The /v2/query/assays/ endpoint delivers metadata collapsed by assay (it always returns id.accession and id.assay name), ideal for quickly locating which assays exist under a set of conditions. [visualization.osdr.nasa.gov](https://visualization.osdr.nasa.gov)

### End-to-end flow (NL → actionable results)

#### 1) Natural-language input

The researcher types a free request, for example:

- “Mouse or human RNA-seq under spaceflight conditions”
- “Proteomics in *Arabidopsis*, any flight/ground annotation”
- “Open OSD-47 RNA-seq”

#### 2) Semantic interpretation (LLM → parameters)

We apply a deterministic prompt system (temperature 0) that converts the sentence into a minimal-filters JSON object:

- **organism** (canonical scientific name, e.g., *Mus musculus*, *Homo sapiens*).



- **condition** (spaceflight | ground | any) with normalization of synonyms (“in-flight,” “ground control,” etc.).
- **assay** (regex-ready fragment, e.g., rna-sequencing|dna-microarray), when applicable.
- **technology** (canonical technology label if the user specifies it: “RNA Sequencing,” “Proteomics,” etc.).
- **dataset** (exact accession if written, e.g., “OSD-47”).

This translation avoids inventing fields not supported by OSDR and leverages regex and OR pipes when appropriate; the backend will wrap the fragments in the semantics of OSDR’s query interface.

### 3) Building the OSDR query

A URL is generated to the assay-grouped endpoint:

- **Endpoint:** /v2/query/assays/
- **Format:** format=json.records (table converted to row-wise JSON)

**Output selectors** (in addition to the two mandatory ones):

- investigation.study assays.study assay technology type (assay technology)
- study.characteristics.organism (organism)
- study.factor value.spaceflight (condition; can also be filtered with regex)
- study.characteristics (full block to extract tissue/organ across several alternative keys)

**Filters** (only if the user expressed them):

- study.characteristics.organism = Mus musculus|Homo sapiens
- investigation.study assays.study assay technology type = RNA Sequencing (RNA-Seq)|Proteomics
- study.factor value.spaceflight

The response includes **applied\_url** to audit the actual query executed (just open it in the browser and OSDR returns the table). The “assays” endpoint collapses duplicates at the assay level and, by design, always prefixes id.accession and id.assay name.

### 4) Row normalization

For each returned record:

- **Organism:** canonical cleanup.
- **Condition:** fine mapping (Pre-/In-/Post-flight) and collapse to **Spaceflight** or **Ground/Analog** for grouping; aliases are detected with robust regexes.



- **Tissue/organ:** attempt extraction from multiple columns (organism part, tissue, organ, cell type, material type, including URL-encoded variants).
- **Technology:** normalization to canonical labels.
- **Links:**
  - **Assay-metadata HTML** (query to `/v2/query/metadata/` with parameters to open a navigable ISA view)
  - **Dataset HTML** (`/v2/dataset/<OSD-###>/?format=html`) for quick inspection.

### 5) De-duplication and Grouping by Technology

Cards are de-duplicated by key (dataset, assay\_name) and aggregated by assay technology.

The final response can:

- List raw (all cards), or
- Group by technology with a top-K limit per group (group\_by\_technology, limit\_per\_tech).

In addition, comparability flags are marked:

- has\_flight, has\_ground, has\_both\_flight\_ground — useful for prioritizing assays with ground controls or flight-ground pairs already available.

### 6) Output Focused on the Researcher's Work

Each card contains:

- dataset (OSDR accession), assay\_name (canonical assay name),
- organism, assay\_technology, conditions (normalized Spaceflight/Ground),
- link (Assay metadata HTML) and dataset\_link (dataset page).

Each technology group includes:

- technology (canonical label),
- count (total assays in the group),
- assays (up to K prioritized cards).

With this, a bioinformatician can quickly prioritize, for example, “*RNA-Seq with flight-ground comparables*” and click into ISA details to assess analytical usability (sample sizes, factor metadata, etc.).



## Appendix C. Implementation details of the Gap Finder (Investor functionality)

### Purpose

**Gap Finder** is built specifically for **managers and investors** who need to detect, justify, and prioritize funding opportunities in spatial biology.

The tool identifies **experimental gaps** where there is no evidence for key combinations of organism, tissue, flight/ground condition, and assay type, and returns an **explainable ranking** with **traceable evidence** linked to real OSDR datasets.

This enables decisions on **where to invest** to maximize the **scientific and strategic return** of new studies.

### What We Consider a “Gap”

A *gap* is the **absence of datasets** for a coarse (aggregated) combination across four dimensions:

$\langle \text{organism}, \text{tissue (normalized to a “parent tissue”)}, \text{condition (Spaceflight | Ground/Analog)}, \text{assay type} \rangle$

A combination is considered **covered** if there is  $\geq 1$  dataset (OSDR accession) observing it (configurable threshold).

This design:

- Is sufficient for most portfolio decisions in the context of the challenge.
- Is expandable to more dimensions (e.g., mission, platform, fine-grained phase as a primary dimension) when appropriate.

### End-to-End Flow (From Free Text to Prioritized Opportunities)

#### 1) Input: Natural Language (Designed for Managers)

The interface offers a **single text field (q)**.

The user writes what they want, e.g.:

- “Opportunities in mouse and human for RNA-Seq and proteomics, any condition.”
- “Spaceflight gaps in mouse liver, prioritize proteomics.”

No need to know the OSDR API or internal vocabularies.

#### 2) Semantic Interpretation (LLM → Structured Filters)

A deterministic prompt system converts *q* into a **filter block**:

- **organisms:** list of scientific names (e.g., *Mus musculus*, *Homo sapiens*)
- **assays:** OSDR technologies (e.g., “RNA Sequencing (RNA-Seq)”, “Proteomics”)
- **condition:** “Spaceflight”, “Ground/Analog”, or “Both”



- **tissues:** list of tissues (normalized text)

#### Key normalizations:

- **Organisms:** robust mapping from common names to Latin; if unspecified, left open.
- **Condition:** collapsed to Spaceflight/Ground-Analog when the intent doesn't distinguish fine phases.
- **Assays:** canonical OSDR labels.
- **Tissues:** "parent tissue" (remove simple plurals, hyphens; Title Case).

If a field doesn't appear in q, it isn't fixed (we maximize the search space).

### 3) Data Extraction (OSDR, json.records Format)

We build and execute a query to:  
/biodata/api/v2/query/assays/?format=json.records

#### Filters (if applicable):

study.characteristics.organism = "Mus musculus|Homo sapiens"

investigation.study\_assays.study\_assay\_technology\_type = "RNA Sequencing (RNA-Seq)|Proteomics"

study.factor\_value.spaceflight = /(space.\*flight|pre.\*flight|post.\*flight|in[- ]?flight)/i

#### Ground/Analog:

/(ground|analog|vivarium|control)/i

For "Both", we don't filter by value; we only select the column (avoids ambiguity).

**Selected columns (always):**  
accession, assay\_name, technology, organism, condition, and the full characteristics block (to extract tissue from alternative keys).

The response includes **applied\_url**, a direct link to the executed query — ensuring **auditability** for managers, investors, and scientists.

### 4) Record Normalization

For each retrieved row:

- **Organism & assay type:** cleaning and standardization.
- **Fine condition (condition\_norm):** may include "Pre-flight", "In-flight", "Post-flight", plus "Spaceflight" and "Ground/Analog".
- **Coarse condition (condition\_coarse):** collapsed to {Spaceflight, Ground/Analog}.
- **Tissue:** search across multiple fields (organism part, tissue, organ, cell type, material type, or URL-encoded variants).



- **Parent tissue:** heuristic (lowercase, remove simple plurals and hyphens; Title Case).
- **OSDR accession:** e.g., OSD-137.

We also build **auxiliary indices** for prioritization:

- Datasets per  $\langle \text{organism, parent\_tissue, condition\_coarse} \rangle$  (activity/neighborhood and Ground baseline).
- Assays present per  $\langle \text{organism, parent\_tissue, condition\_coarse} \rangle$  (multi-omics).
- Fine phases per  $\langle \text{organism, parent\_tissue} \rangle$  (phase criticality).
- Species that cover  $\langle \text{parent\_tissue, condition\_coarse, assay} \rangle$  (translation).

## 5) Coverage Calculation and Base Gap Detection

**Scope (evaluation universe)** = Cartesian product of the fixed axes (or of observed values if no filters).

For each key  $\langle \text{organism, parent\_tissue, condition\_coarse, assay\_type} \rangle$ , we count **unique datasets**.

A key is a **gap** if there are **no datasets** (or fewer than the threshold `min_datasets_for_covered`, default 1).

This base set is **objective and reproducible** — from here we apply a **justifiable ranking**.

## 6) Prioritization (Score) and Explainability (Reasons)

For each gap we compute **signals (0–1)**, combined into a weighted score:

Signal	Description
<b>GroundBase</b>	Strength of the Ground/Analog base for same organism/tissue (capped at 3).
<b>MultiOmics</b>	High if a complementary layer exists (e.g., RNA-Seq present, Proteomics missing).
<b>PhaseCritical</b>	High if Pre/Post exist but In-flight missing (or vice versa).
<b>SpeciesTranslation</b>	High if combination exists in another species ( <i>Mus</i> $\leftrightarrow$ <i>Homo</i> ).
<b>NeighborDensity</b>	Total activity in the neighborhood (proxy for feasibility).
<b>Feasibility</b>	Relative frequency of the assay in the scope (maturity).
<b>Redundancy (penalty)</b>	Prevents Top-N from filling with near-duplicates.

**Example weighted score:**

$$\text{score} = 1.8 \cdot \text{GroundBase} + 1.5 \cdot \text{MultiOmics} + 1.2 \cdot \text{PhaseCritical}$$





+ 1.0·SpeciesTranslation + 0.8·NeighborDensity

+ 0.6·Feasibility – 0.7·Redundancy

### Explainability (“Reasons”)

- **reason (primary):** main contributing signal (e.g., “Strong ground base and missing in flight”).
- **reasons\_detail[]:** supporting motives including:
  - type (GroundBase, MultiOmics, etc.)
  - normalized\_value (0–1)
  - explanation (natural language)
  - evidence (when applicable), such as:
    - OSDR links (accessions)
    - Present/missing layers (multi-omics)
    - Detected phases (phase criticality)
    - Species with coverage (translation)

The result is a **Top-K list** of opportunities that both **scores** and **justifies** where to invest — and **why**.

### API Output (Consumable by the UI and Auditors)

GET /gaps/search?q=<text>&top\_n=<K>&min\_datasets\_for\_covered=<m>

#### Returns:

- applied\_url: exact OSDR query executed (clickable).
- highlights: prioritized list of gaps, each with:
  - organism, tissue (parent), condition (coarse), assay\_type
  - score, reason, reasons\_detail[] (with evidence and OSDR links)
- gaps\_total: total number of gaps in scope.
- gaps: full listing (for table/export).



## Appendix D. Implementation details of the Mission Planner (Mission Architect functionality)

### Problem and Opportunity

#### Problem:

Early-stage mission architecture definition requires synthesizing a large volume of evidence and prior experience (analog missions) into **consistent, evaluable proposals**.

#### Opportunity:

An expert assistant with **structured output** reduces analysis cycles, improves traceability, and enables automated dashboards, scoring, and comparative assessments.

### Solution Overview

#### Flow

1. Receives user prompt + search method (local / global / drift).
2. Executes **GraphRAG** and persists the exchange in the chat.
3. Re-contextualizes the response with a system prompt “**AstroPlan**” enforcing **strict JSON output**.
4. Validates JSON and returns it to the client.

#### Implementation References:

- post\_chat\_astroplan: api/graphbot/chats/router.py:43
- Chat logic: api/graphbot/chats/service.py:21
- GraphRAG engine: api/graphbot/chats/chatbot/graphrag\_bot.py:13
- OpenAI provider: api/ai/providers/openai\_provider.py:19

### Technical Architecture

#### API Layer

FastAPI router in api/graphbot/chats/router.py:43.

#### Chat Service

ChatService.reply\_to\_user handles **locking by chat\_uuid** and **history persistence** (api/graphbot/chats/service.py:21).

#### Search Engine

**GraphRAG** (local / global / drift) offloaded using `asyncio.to_thread` (api/graphbot/chats/chatbot/graphrag\_bot.py:15).

#### LLM Provider

**OpenAI GPT-4o**, with controlled parameters (temperature=0.2, max\_tokens=2000) (api/ai/providers/openai\_provider.py:25).



## Configuration

- GRAPHRAG\_ROOT and in-memory store (api/graphbot/settings.py:6, api/graphbot/factory.py:10)
- CORS & dependency wiring: api/app.py:34, api/app.py:56

## Interface (I/O)

### Request

#### Schema:

MessageRequest

#### Fields:

- message (user text) — api/graphbot/chats/schemas.py:8
- metodo (“local” | “global” | “drift”) — api/graphbot/chats/schemas.py:10

### Errors

- **409** – Chat locked/busy (api/graphbot/chats/service.py:14, router.py:131)
- **502** – Invalid JSON returned by OpenAI (router.py:120)

## Execution Flow

1. **ChatService.reply\_to\_user** executes GraphRAG and appends messages to chat history (service.py:32).
2. Builds **system prompt** and **user input** using GraphRAG context (router.py:60, router.py:101).
3. Calls **OpenAIProvider.prompt** with GPT-4o (router.py:112, openai\_provider.py:25).
4. Performs **json.loads validation** and returns a **JSONResponse** (router.py:120).

## Example Usage

### Endpoint:

POST /api/v1/chats/{chat\_uuid}/astroplan

### Body:

```
{
  "message": "Architecture for a Phobos mission in 2030",
  "metodo": "global"
}
```

**Response:**

A strict JSON object conforming to the AstroPlan schema, directly consumable by dashboards.