# Simple RAG System for SpaceX Projects

The project Objective's

Develop a basic Retrieval-Augmented Generation (RAG) system using a diverse dataset to answer questions specifically related to SpaceX.

The primary goal of this project is to build a RAG system that can respond to user queries about SpaceX, their ongoing projects, and their latest space missions by utilizing a combination of different data sources.

The project consists of 2 main parts the first part of the project discusses the data sets and information resources the second part of the project forms the pipeline of the project

## Part (1) Dataset Acquisition and Processing

In this project, various datasets are collected from multiple resources, processed, and stored in my Google Drive for easy access and use. The collected data focuses on providing comprehensive information relevant to SpaceX and its activities.

### Data Sources

The dataset for this RAG system is derived from four main types of sources, each contributing unique information to form a complete and informative resource:

- **PDF Files**: These include SpaceX mission guides, general information about SpaceX's rocket fleet, and detailed documentation about the company's history, vision, and mission. The PDFs are preprocessed, split into smaller chunks, and stored as text files.
- **Wikipedia Articles**: Articles sourced from Wikipedia provide scientific insights and background on topics related to space exploration, space vehicles, rocket technology, and historical space missions.
- **CC News Articles**: This data source comprises journalistic articles covering SpaceX, NASA, and other prominent space organizations. These articles highlight key achievements, collaborations, and the impact of SpaceX on the space industry.
- **Video Transcripts**: Videos related to space exploration, SpaceX's ambitious goals, and the intricacies of rocket engines are transcribed. The transcripts offer a detailed, comprehensive view of the company's work and its future vision for space travel and technology.

### Preprocessing and Storage

After collecting data from the above sources, each resource was preprocessed using text splitting, cleaning, and formatting techniques. The processed data was then stored in .txt files for seamless retrieval and integration into the RAG system.

Purpose and Use Case

The resulting RAG system is designed to serve as a Q&A service, focusing on providing accurate and contextual information regarding SpaceX's endeavors, current projects, and their broader implications for space exploration. By leveraging this system, users can gain deeper insights into the company's efforts and keep up-to-date with their latest achievements and missions.

## Part (2) Implement the RAG Pipeline

Objective

Develop a basic Retrieval-Augmented Generation (RAG) system using different LLM models, embeddings, and libraries like LangChain to answer questions specifically related to SpaceX.

The purpose of this step is to integrate various components such as data retrieval, embedding, and language models to build a robust RAG system that can handle queries efficiently.

Dataset Chunking

In this step, datasets from different resources are collected and processed to create manageable chunks of text. The chunking process is tailored according to the requirements of the system, ensuring that each chunk is small enough for efficient retrieval but large enough to maintain context.

Resources Contributed:

- **PDFs**: Mission guides, technical documents, and SpaceX fleet overviews.
- **Wikipedia Articles**: Informative content about space exploration and rocket technology.
- **News Articles**: Recent updates and journalism focusing on SpaceX's achievements.
- **Video Transcripts**: Video content covering SpaceX's technology and innovations.

Embedding the Chunks into Vectors

After chunking the dataset, the text chunks are converted into vector representations. In this step, I used the models/text-embedding-004 from **Gemini embeddings** due to its **high performance in capturing semantic relationships**. The model is chosen for its:

- Ability to represent complex sentences accurately.
- Lightweight nature, making it suitable for smaller-scale applications.
- Good balance between computational cost and embedding quality.

Data Vectorizing

To vectorize the data and build the vector database, I used **Chroma** for its **flexibility and ease of integration**. Chroma offers efficient vector storage and retrieval capabilities, making it a suitable choice for a retrieval-based pipeline. Its key features include:

- **Scalability**: Suitable for handling larger datasets.
- **Performance**: Optimized for fast vector searches.

LLM Models

In this project, two models from **Gemini** were utilized: gemini_pro 1.5 and gemini_pro_flash 1.5. The primary difference between the two is their **speed and memory efficiency**:

- **Gemini Pro 1.5**: Offers high-quality generation and a deeper understanding of complex queries but might be slower for real-time applications.
- **Gemini Pro Flash 1.5**: Prioritizes speed and resource efficiency, making it ideal for scenarios where quick responses are required, such as chatbots or real-time query handling.

The user can select between these models depending on the trade-off between **response quality and speed**.