### Project Problem Statement:

Creating a comprehensive data collection, analysis, and visualization platform for YouTube content is a challenging task. The problem involves gathering data from YouTube, storing it in a structured manner, and providing robust mechanisms to query and analyze the data.

### Project Overview:

The goal of this project is to build a data harvesting and warehousing system for YouTube that allows users to extract data from YouTube, store it in a database, and query the stored data to gain insights about channels, videos, comments, and playlists. This data platform should:

* Enable users to input YouTube channel IDs to retrieve information.
* Store channel, video, comment, and playlist data in a MongoDB database.
* Transfer the stored data to PostgreSQL for SQL-based querying and analytics.
* Provide an interface for users to interact with the platform and query the data for insights.

### Explanation of the Project:

#### Data Collection:

* **YouTube Data API:** The project uses the YouTube Data API to fetch data about channels, videos, comments, and playlists. Given a channel ID, the system collects various details about the channel, including subscriber count, view count, video count, and playlist details.
* **Functions to Fetch Data:** Functions are defined to collect information about specific channels, videos, comments, and playlists. These functions interact with the YouTube Data API and retrieve the required information.

#### Data Storage and Warehousing:

* **MongoDB Storage:** MongoDB is used to store the harvested data. This is useful because MongoDB's flexible schema allows for storing various types of information without a strict structure, making it easy to adapt to changes in the data format.
* **PostgreSQL Storage:** PostgreSQL is used for warehousing the data. This allows SQL-based querying, which is highly efficient for complex analytics and aggregations. The data is transferred from MongoDB to PostgreSQL for querying purposes.

#### Data Retrieval and Display:

* **Streamlit Interface:** Streamlit is used to create an interactive web interface. This allows users to input channel IDs, fetch data, and view stored data in a user-friendly manner.
* **User Interaction:** The Streamlit interface provides various features, such as text input for channel IDs, buttons to trigger data collection, dropdowns to select channels for migration, and radio buttons to choose which tables to view.
* **SQL Queries for Analysis:** The PostgreSQL database contains structured data, enabling complex SQL queries. The project provides SQL-based analytics to answer common questions, such as:
  + What are the 10 most viewed videos?
  + Which channels have the most videos?
  + What is the average duration of videos in each channel?
  + Which videos have the highest number of likes or comments?
  + Which videos were published in a specific year?
  + How many comments are there for each video?

### Overall Architecture:

1. **Data Collection from YouTube:** Retrieve data from the YouTube Data API using specified channel IDs.
2. **Storage in MongoDB:** Store the collected data in MongoDB for flexibility and scalability.
3. **Migration to PostgreSQL:** Transfer data from MongoDB to PostgreSQL for SQL-based querying and analysis.
4. **User Interface with Streamlit:** Provide an interface for users to interact with the system, collect data, and query it for insights.

### Challenges and Considerations:

* **API Rate Limits:** Be mindful of the YouTube Data API's rate limits when making API calls.
* **Data Consistency and Structure:** Ensure that data is consistent and structured properly in the database.
* **Efficient Queries:** Design efficient SQL queries for rapid data retrieval and analysis.
* **User Experience:** Provide a user-friendly interface with clear feedback and intuitive controls.

### Conclusion:

This project creates a comprehensive platform for data harvesting and warehousing of YouTube content. By leveraging both MongoDB and PostgreSQL, the system provides flexible data storage and powerful SQL-based analytics. The Streamlit interface makes it easy for users to interact with the system and gain insights from YouTube data.