**Introduction:**

"YouTube Data Harvesting and Warehousing using SQL, MongoDB and Streamlit in Data Science" project, I am thrilled to delve into the realm of extracting, storing, and visualizing insights from one of the most influential platforms on the internet: YouTube

In this project, I will explore the vast landscape of YouTube data, tapping into its rich repository of videos, engagement metrics, and audience behaviors. By harnessing the power of APIs and data harvesting techniques, I aim to collect a diverse range of data points, including views, likes, comments, and user interactions.

**Data Collection**: Utilize YouTube's API to gather a comprehensive dataset including video metrics such as views, likes, comments, and user engagement data.

**Data Transformation**: Process and transform the harvested data to fit the schemas of both SQL and MongoDB databases, ensuring compatibility and efficiency in storage.

**Data Warehousing**: Implement storage solutions using SQL and MongoDB databases to organize and manage the harvested YouTube data effectively. SQL databases will provide structured storage for relational analysis, while MongoDB will offer flexibility for unstructured or semi-structured data.

**Dashboard Creation:** Develop an interactive dashboard using Streamlit to visualize the YouTube data in a user-friendly and accessible manner. The dashboard will allow stakeholders to explore insights, trends, and patterns within the data through dynamic visualizations and custom queries.

**Insight Generation:** Analyze the stored data to extract meaningful insights into YouTube content performance, audience engagement, and emerging trends. These insights will inform content creators, marketers, and decision-makers in optimizing their strategies for success on the platform.

**Learning and Skill Development**: Gain hands-on experience in data harvesting, warehousing, and visualization technologies, including APIs, SQL, MongoDB, and Streamlit. Develop proficiency in data science methodologies and tools through practical application in a real-world project scenario.

**REQUIRED LIBRARIES:**

* googleapiclient.discovery
* streamlit
* psycopg2
* pymongo
* pandas

**Objective:** Develop a Streamlit application for accessing and analyzing data from multiple YouTube channels.

***Features:***

* Retrieve channel details and video data using YouTube API.
* Store data in a data lake.
* Option to store data in Mongodb, MySQL or PostgreSQL.
* Ability to search and retrieve data from the SQL database.

**Approach Overview**

* Set up a Streamlit app for user interface.
* Connect to the YouTube API using Google API client library.
* Store and clean data temporarily using pandas DataFrames.
* Migrate data to a SQL data warehouse (MySQL or PostgreSQL).
* Query the SQL data warehouse with SQL queries.
* Display data in the Streamlit app using data visualization features.

**Streamlit App Setup**

Explanation: Streamlit chosen for rapid UI development.

***Features***

* User input for YouTube channel ID.
* Display channel details.
* Option to select channels for migration.

**Connecting to YouTube API**

* Utilizing Google API client library for Python.
* Making requests to the YouTube API to retrieve channel and video data.
* Discuss authentication and quota considerations.

**Data Storage and Cleaning**

* Storing retrieved data temporarily using pandas DataFrames.
* Cleaning and preprocessing data for migration.
* Ensuring data integrity and consistency.

**Migration to SQL Data Warehouse**

* Explanation: Data migration to SQL databases (MySQL or PostgreSQL).
* Benefits of using SQL data warehouse for structured data storage.
* Discuss schema design considerations.

**Querying the SQL Data Warehouse**

* Using SQL queries to retrieve data for specific channels.
* Joining tables to get comprehensive channel details.
* Integrating Python SQL library like SQLAlchemy for interaction.

**Displaying Data in Streamlit**

* Utilizing Streamlit's data visualization features.
* Creating charts and graphs for data analysis.
* Providing interactive elements for user engagement.