**Project Title: YouTube Data Harvesting and Warehousing**

**Project Overview:**

The project involves collecting data from YouTube using the YouTube Data API, storing the collected data in both MongoDB and MySQL databases, and providing an interface using Streamlit to view and analyze the collected data.

**Workflow:**

1. **API Connection and Data Retrieval:**
   * Use the Google API Python Client to connect to the YouTube Data API.
   * Implement functions to retrieve **channel information, playlist details, video information, and comments** from the YouTube channel.
   * Store the retrieved data in **MongoDB**.
2. **Data Warehousing in MySQL:**
   * Define functions to create tables in a MySQL database for storing channel details, playlist information, video details, and comments.
   * Transfer the data from **MongoDB** to **MySQL** tables.
3. **Streamlit Interface:**
   * Implement a Streamlit application to provide a user interface for interacting with the stored data.
   * Create functions to display the data from the MySQL tables in the Streamlit app.
   * Allow users to trigger the data collection process for required YouTube channel, store it in MongoDB, and transfer it to MySQL.
4. **Data Analysis Queries:**
   * Write SQL queries to analyze the stored data for various insights.
   * Implement functionality in the Streamlit app to execute these queries based on user selection.
   * Display the query results in the Streamlit app for user analysis.

**Execution:**

1. **Setting Up Environment:**
   * Install required Python packages such as **google-api-python-client**, **isodate**, **pandas**, **pymongo**, **mysql-connector-python**, and **streamlit**.
2. **API Key Configuration:**
   * Obtain a YouTube Data API key and configure it in the Python script for API access.
3. **Running the Application:**
   * Execute the Python script containing the Streamlit application.
   * The application will provide options to collect data from YouTube, store it in MongoDB, transfer it to MySQL, and view the data through different queries in the Streamlit interface.
4. **Interacting with the Streamlit App:**
   * Users can select various options such as viewing channel details, playlist information, video details, comments, and executing specific SQL queries for data analysis.
   * The Streamlit app will display the requested data or query results, providing users with insights into the collected YouTube data.

By following this workflow and executing the provided Python script, users can effectively collect, store, analyze, and visualize data from YouTube using MongoDB, MySQL, and Streamlit.