# **YouTube Data Harvesting and Warehousing Project**

This project aims to collect comprehensive data from YouTube channels, videos, playlists, and comments using the YouTube Data API. The collected data is stored in a SQLite3 database and visualized interactively using Plotly within a Streamlit application. Users can query the stored data to gain insights into channel performance, video engagement metrics, and audience interaction.

## **Technologies Used**

**Python**: The primary programming language used for implementing the project, including data collection, analysis, and database management.

**YouTube Data API**: Google's official API leveraged to interact with YouTube's platform for retrieving comprehensive data about channels, videos, playlists, and comments.

**SQLite3**: A lightweight, serverless database engine chosen for its simplicity and ease of integration with Python, used for storing structured data collected from YouTube.

**Streamlit**: An open-source app framework used to create interactive web applications for visualizing and querying the collected YouTube data stored in SQLite3 databases.

**Plotly**: A Python graphing library used for creating interactive visualizations in the Streamlit application to provide insights into channel and video analytics.

## **Usage**

1. **Data Collection**:
   * Enter a YouTube channel ID to fetch and store comprehensive data about the channel, including statistics, videos, playlists, and comments.
   * Data collected includes channel statistics (subscribers, views, total videos), video details (title, description, views, likes, dislikes), playlist details, and video comments.
2. **Data Storage**:
   * Utilizes SQLite3 database to store structured data:
     + **Channels**: Stores detailed information about YouTube channels.
     + **Videos**: Stores specific data about each video including views, likes, dislikes, and comments.
     + **Playlists**: Records playlist details associated with channels.
     + **Comments**: Stores comments retrieved from YouTube videos, including author details and timestamps.
3. **Visualization and Querying**:
   * Provides interactive visualizations using Plotly within the Streamlit application:
     + Channel statistics (subscribers, views).
     + Video details (views, likes, dislikes).
     + Playlist details (title, video count).
     + Comment details (text, author).
   * Supports querying stored data with predefined SQL queries to generate reports:
     + Most viewed videos and channels.
     + Top videos by likes, comments, and views.
     + Average video duration by channel.
     + Channels with videos published in specific years.

## **Features**

1. **Comprehensive Data Retrieval**:
   * Retrieves detailed information about YouTube channels, videos, playlists, and comments using the YouTube Data API.
2. **Efficient Data Storage**:
   * Stores collected data in a SQLite3 database, ensuring easy retrieval and efficient data management.
3. **Interactive Visualization**:
   * Uses Plotly and Streamlit to create interactive visualizations and reports, enabling users to gain insights into YouTube channel performance and video engagement.
4. **Querying Capabilities**:
   * Supports predefined SQL queries to extract specific insights and generate reports based on stored YouTube data.
5. **User-Friendly Interface**:
   * Provides a user-friendly interface through Streamlit for entering channel IDs, visualizing data, and querying stored information.

This README file provides an overview of the YouTube Data Harvesting and Warehousing project, detailing its objectives, technologies used, core functionalities, and usage instructions. Adjustments can be made as necessary to reflect specific project requirements and updates.