YouTube Data Harvesting and Warehousing using SQL, MongoDB and Streamlit

by **Navin Kumar**

TABLE OF CONTENTS

1. INTRODUCTION

Project Snapshot

YouTube Data Harvesting and Warehousing is a project designed to empower users to access and analyze data from various YouTube channels. The application, developed using **Python, SQL, MongoDB, and Streamlit** provides a user-friendly interface for retrieving, saving, and querying YouTube channel and video data.

Technology Used

1 **Python**

As the primary programming language, Python is employed for the complete application development, including data retrieval, processing, analysis, and visualization.

2 **Google API Client**

The googleapiclient library in Python facilitates communication with YouTube's Data API v3, allowing seamless retrieval of essential information like channel details, video specifics, and comments.

3 **MongoDB**

MongoDB, a scalable document database, is used for storing structured or unstructured data in a JSON-like format.

4 **MySQL**

MySQL, an advanced and scalable open-source DBMS, is employed for efficient storage and management of structured data, offering support for various data types and advanced SQL capabilities.

5 **Streamlit**

The Streamlit library is utilized to create an intuitive UI, enabling users to interact with the application for data retrieval and analysis.

Required Python Libraries

1. **googleapiclient.discovery**
2. **streamlit**
3. **sqlalchemy**
4. **pymysql**
5. **pymongo**
6. **pandas**

2. DATA SCRAPING

Retrieving Data From YouTube API

1

Channel Details

The application will allow users to input a YouTube channel ID and retrieve all the relevant data, including channel details such as the **channel's name, description, subscription count, channel views, and upload playlist id.**

2

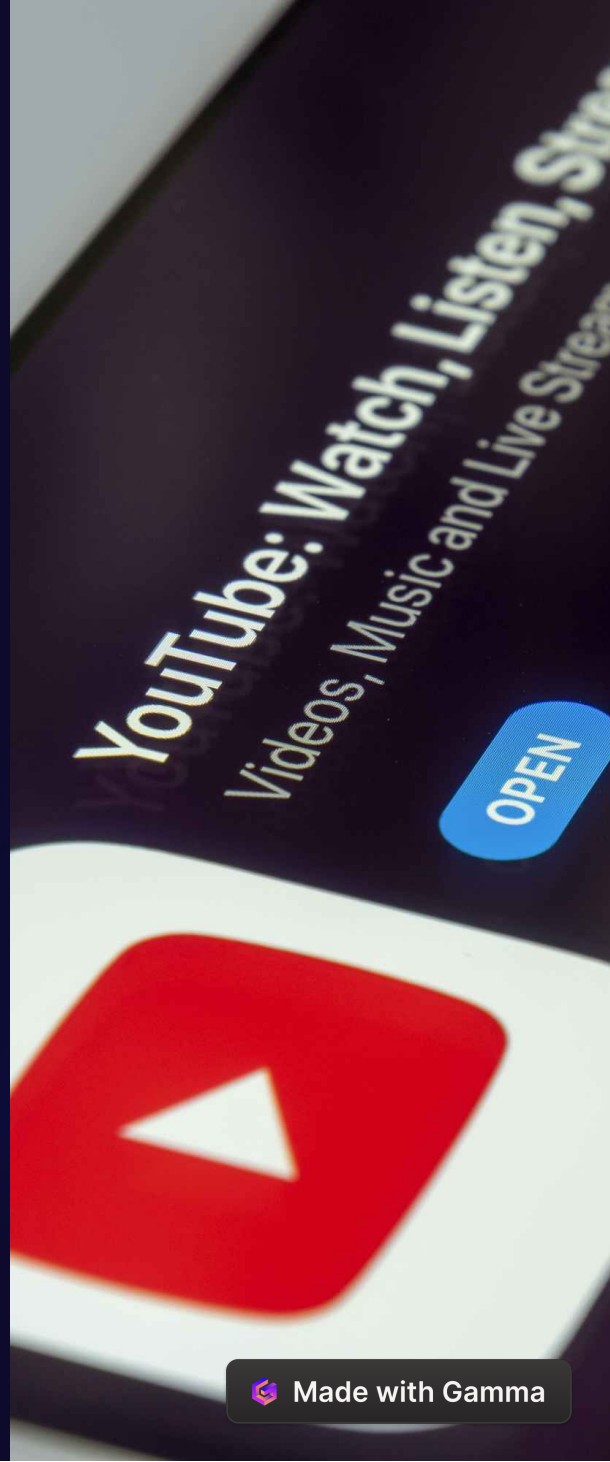
Video Details and Stats

It will also retrieve video details, such as **video title, video id, description, publishing date, thumbnail, caption status and duration** along with statistical data like **number of views, likes, and comments**.

3

Comment Data

Additionally, the application will extract comment data, including **comment content, comment ID, comment author, publish date** using the Google API.



3. DATA STORING

Storing Data in MongoDB Data Lake

Data Storage

The retrieved data from the YouTube API will be stored in a MongoDB database as a data lake, providing a flexible and scalable storage solution.

Schema-less Nature

Allows storing diverse data types without predefined schema.

Scalability

Can easily handle growing datasets and scale as per requirements.

Flexibility

Enables efficient data retrieval and query execution.

Migrating Data to SQL Database

1 Data Transformation

The data from the MongoDB data lake will be transformed into structured data and migrated to a SQL database as tables for efficient querying and analysis.

2 Schema Specification

Each type of data will be mapped to the appropriate database schema to ensure data integrity and optimize performance.

3 Flexible Searches

Find what you need faster! powerful search system quickly finds the information you want in the way you want it.

Analyzing Data From SQL Database

Pre-Written Questions

The Streamlit application will include pre-written questions for data analysis, allowing users to explore and interpret the data from the SQL database.

4. USER INTERFACE

Streamlit Application User Interface

1

Data Access

Users can easily input a YouTube channel ID and access comprehensive data, strengthening their decision-making process.

2

Data Storage

The MongoDB data lake data warehouse ensure secure and organized information storage.

3

Migration to SQL

Migrate the collected YouTube channel data from MongoDB to a SQL database, enabling further analysis and integration with other systems.

4

Analyze

Users can analyze the data with pre-written queries to gain insights into the performance of the YouTube channel, identify trends, and make data-driven decisions to optimize their YouTube strategy.

THANKS