**YouTube Data Harvesting and Warehousing using SQL, MongoDB, and Streamlit**

Project Link

**Introduction**

YouTube Data Harvesting and Warehousing is a project that aims to allow users to access and analyze data from multiple YouTube channels. The project utilizes SQL, MongoDB, and Streamlit to create a user-friendly application that allows users to retrieve, store, and query YouTube channel and video data.

**Project Overview**

The YouTube Data Harvesting and Warehousing project consists of the following components:

* Streamlit Application: A user-friendly UI built using Streamlit library, allowing users to interact with the application and perform data retrieval and analysis tasks.
* YouTube API Integration: Integration with the YouTube API to fetch channel and video data based on the provided channel ID.
* MongoDB Data Lake: Storage of the retrieved data in a MongoDB database, providing a flexible and scalable solution for storing unstructured and semi-structured data.
* SQL Data Warehouse: Migration of data from the data lake to a SQL database, allowing for efficient querying and analysis using SQL queries.
* Data Visualization: Presentation of retrieved data using Streamlit's data visualization features, enabling users to analyze the data through charts and graphs.

**Technologies Used**

The following technologies are used in this project:

* Python: The programming language used for building the application and scripting tasks.
* Streamlit: A Python library used for creating interactive web applications and data visualizations.
* YouTube API: Google API is used to retrieve channel and video data from YouTube.
* MongoDB: A NoSQL database used as a data lake for storing retrieved YouTube data.
* SQL (MySQL): A relational database used as a data warehouse for storing migrated YouTube data.
* SQLAlchemy: A Python library used for SQL database connectivity and interaction.
* Pandas: A data manipulation library used for data processing and analysis.
* Matplotlib: A data visualization library used for creating charts and graphs.

**Installation and Setup**

To run the YouTube Data Harvesting and Warehousing project, follow these steps:

1. Install Python: Install the Python programming language on your machine.
2. Install Required Libraries: Install the necessary Python libraries using pip or conda package manager. Required libraries include Streamlit, MongoDB driver, SQLAlchemy, Pandas, and Matplotlib.
3. Set Up Google API: Set up a Google API project and obtain the necessary API credentials for accessing the YouTube API.
4. Configure Database: Set up a MongoDB database and SQL database (MySQL) for storing the data.
5. Configure Application: Update the configuration file or environment variables with the necessary API credentials and database connection details.
6. Run the Application: Launch the Streamlit application using the command-line interface.

**Usage**

Once the project is setup and running, users can access the Streamlit application through a web browser. The application will provide a user interface where users can perform the following actions:

* Enter a YouTube channel ID to retrieve data for that channel.
* Store the retrieved data in the MongoDB data lake.
* Collect and store data for multiple YouTube channels in the data lake.
* Select a channel and migrate its data from the data lake to the SQL data warehouse.
* Search and retrieve data from the SQL database using various search options.
* Perform data analysis and visualization using the provided features.

**Features**

The YouTube Data Harvesting and Warehousing application offers the following features:

* Retrieval of channel and video data from YouTube using the YouTube API.
* Storage of data in a MongoDB database as a data lake.
* Migration of data from the data lake to a SQL database for efficient querying and analysis.
* Search and retrieval of data from the SQL database using different search options, including joining tables.
* Data analysis and visualization through charts and graphs using Streamlit's data visualization capabilities.
* Support for handling multiple YouTube channels and managing their data.

**Future Enhancements**

Here are some potential future enhancements for the YouTube Data Harvesting and Warehousing project:

* Authentication and User Management: Implement user authentication and management functionality to secure access to the application.
* Scheduled Data Harvesting: Set up automated data harvesting for selected YouTube channels at regular intervals.
* Advanced Search and Filtering: Enhance the search functionality to allow for more advanced search criteria and filtering options.
* Additional Data Sources: Extend the project to support data retrieval from other social media platforms or streaming services.
* Advanced-Data Analysis: Incorporate advanced analytics techniques and machine learning algorithms for deeper insights into YouTube data.
* Export and Reporting: Add features to export data and generate reports in various formats for further analysis and sharing.

**Conclusion**

The YouTube Data Harvesting and Warehousing project provides a powerful tool for retrieving, storing, and analyzing YouTube channel and video data. By leveraging SQL, MongoDB, and Streamlit, users can easily access and manipulate YouTube data in a user-friendly interface. The project offers flexibility, scalability, and data visualization capabilities, empowering users to gain insights from the vast amount of YouTube data available.

**References**

* Streamlit Documentation: <https://docs.streamlit.io/>
* YouTube API Documentation: <https://developers.google.com/youtube>
* MongoDB Documentation: <https://docs.mongodb.com/>
* SQLAlchemy Documentation: <https://docs.sqlalchemy.org/>
* Python Documentation: <https://docs.python.org/>
* Matplotlib Documentation: <https://matplotlib.org/>