**Project Name:**

YouTube Data Harvesting and Warehousing using SQL, MongoDB and Streamlit

**Skills take away From This Project:**

Python scripting, Data Collection, MongoDB, Streamlit, API integration, Data Management using MongoDB (Atlas) and SQL

**Problem Statement:**

The problem statement is to create a Streamlit application that allows users to access and analyze data from multiple YouTube channels. The application should have the following features: 1. Ability to input a YouTube channel ID and retrieve all the relevant data (Channel name, subscribers, total video count, playlist ID, video ID, likes, dislikes, comments of each video) using Google API. 2. Option to store the data in a MongoDB database as a data lake. 3. Ability to collect data for up to 10 different YouTube channels and store them in the data lake by clicking a button. 4. Option to select a channel name and migrate its data from the data lake to a SQL database as tables. 5. Ability to search and retrieve data from the SQL database using different search options, including joining tables to get channel details.

**Route Map:**

1. Set up a Streamlit app: Streamlit is a great choice for building data visualization and analysis tools quickly and easily. You can use Streamlit to create a simple UI where users can enter a YouTube channel ID, view the channel details, and select channels to migrate to the data warehouse.

2. Connect to the YouTube API: You'll need to use the YouTube API to retrieve channel and video data. You can use the Google API client library for Python to make requests to the API.

3. Store data in a MongoDB data lake: Once you retrieve the data from the YouTube API, you can store it in a MongoDB data lake. MongoDB is a great choice for a data lake because it can handle unstructured and semi-structured data easily.

4. Migrate data to a SQL data warehouse: After you've collected data for multiple channels, you can migrate it to a SQL data warehouse. You can use a SQL database such as MySQL or PostgreSQL for this.

5. Query the SQL data warehouse: You can use SQL queries to join the tables in the SQL data warehouse and retrieve data for specific channels based on user input. You can use a Python SQL library such as SQLAlchemy to interact with the SQL database.

6. Display data in the Streamlit app: Finally, you can display the retrieved data in the Streamlit app. You can use Streamlit's data visualization features to create charts and graphs to help users analyze the data. Overall, this approach involves building a simple UI with Streamlit, retrieving data from the YouTube API, storing it in a MongoDB data lake, migrating it to a SQL data warehouse, querying the data warehouse with SQL, and displaying the data in the Stream lit app.

**Output:**

Based on the retrieved data we can able to analyse/visualize the each object. Can able to focus o the area of improvement