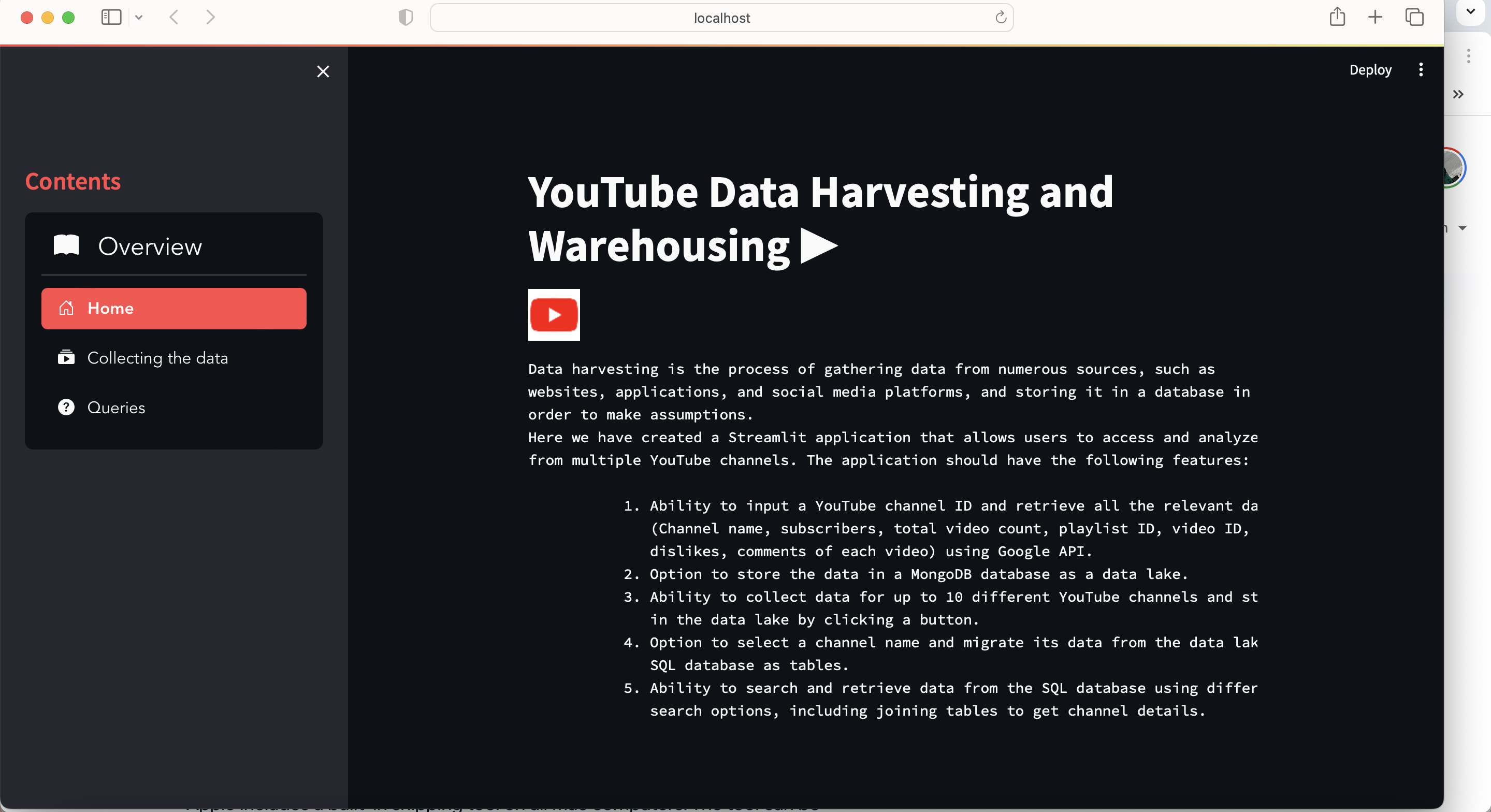
**YouTube Data Harvesting and Warehousing.**



**Description:**

**•** YouTube has become a powerful tool for individuals, creators, and businesses to share their stories, express themselves, and connect with audiences worldwide.

• The "YouTube Data Harvesting” is a Streamlit application designed to provide users with a platform to access and analyze data from multiple YouTube channels. The application is equipped with several features, enabling users to input a YouTube channel ID and retrieve relevant data, store data in a MongoDB database as a data lake, migrate data to a SQL database, and search and retrieve data from the SQL database.

**User Guide:**

**Prerequisites**

* Python 3.7 or later
* Anaconda-Navigator(VSCode)
* YouTube Data API key
* MongoDB installed and running
* SQL database (e.g.,MySQLWorkbench) installed and running

**Installation Instructions:**

* pip install google-api-python-client, pymongo, mysql-connector-python, pymysql, pandas, streamlit.

( pip install google-api-python-client pymongo mysql-connector-python pymysql pandas streamlit )

(install google-api-python-client pymongo mysql-connector-python sqlalchemy pymysql pandas numpy plotly-express streamlit )

**Import Libraries**

**Youtube API libraries**

• import googleapiclient.discovery

• from googleapiclient.discovery import build

**MongoDB**

• import pymongo

**SQL libraries**

• import mysql.connector

• import pymysql

**pandas, numpy**

• import pandas as pd

**Dashboard libraries**

• import streamlit as st

**ETL :**

When designing an ETL process for harvesting data from YouTube, you'll need to consider the specific data you want to extract, how you'll transform it to fit your needs, and where you'll load the data. Here's an example ETL process for harvesting data from YouTube:

**Extract:**

• YouTube Data API:

• Use the YouTube Data API to extract relevant data such as video details, channel information, and statistics.

• Authenticate your requests using an API key.

**Transform:**

• Data Cleaning and Transformation:

• Handle missing or inconsistent data by cleaning it during the transformation phase.

• Convert data types and formats to meet your desired schema.

• Enrich the data by adding calculated fields or additional information.

• Data Aggregation:

• Summarize statistics such as likes, dislikes, and comments for each video or channel.

**Load:**

**Data Storage:**

• **MongoDB Data Lake:** Store raw or semi-processed data in MongoDB as a data lake for flexibility.

• **SQL Database:** Load aggregated and transformed data into an SQL database for structured storage.

• **Incremental Loading:** Implement mechanisms for incremental loading to update the database with new data periodically.Avoid reloading all data each time to optimize performance.

**Exploratory Data Analysis (EDA) Framework:**

**a) Access MySQL DB**

• Create a connection to the MySQL server and access the specified MySQL DataBase by using pymysql library and access tables.

**b) Filter the data**

• Filter and process the collected data from the tables depending on the given requirements by using SQL queries and transform the processed data into a DataFrame format.

**c) Visualization**

• Finally, create a Dashboard by using Streamlit and give dropdown options on the Dashboard to the user and select a question from that menu to analyse the data and show the output in Dataframe Table and Bar chart.

**User Guide**

**Step 1. Data collection zone**

• Search channel\_id, copy and paste on the input box and click the Get data and stored button in the Data collection zone.

**Step 2. Data Migrate zone**

• Select the channel name and click the Migrate to MySQL button to migrate the specific channel data to the MySQL database from MongoDB in the Data Migrate zone.

**Step 3. Channel Data Analysis zone**

• Select a Question from the dropdown option you can get the results in Dataframe format or bar chat format.

