CAPSTONE 1: YOUTUBE DATA HARVESTING AND WAREHOUSING USING SQL AND STREAMLIT

1. *Setup Environment:*

- Create a Python virtual environment for the project.

Install Streamlit using Anaconda Distribution - Streamlit Docs

- Install necessary packages like Streamlit, Pandas, MySQL Connector, and Google API client library.

2. *Google API Integration:*

- Obtain API credentials from the Google Developers Console for accessing the YouTube Data API.

https://colab.research.google.com/drive/10PKu9YvhoPyIeWEjIAVuoJrVb3-snU8I?usp=sharing#scrollTo=SKOwdi9QdbJ9

- Use the Google API client library to interact with the YouTube Data API

- Implement functionality to extract information on a YouTube channel, such as videos, statistics, and metadata.

3. *Database Setup:*

- Set up a MySQL database to store the extracted information.

- Design the database schema to represent YouTube channel data effectively.

- Create tables for storing channel details, video information, and any other relevant data.

4. *Data Extraction and Storage:*

- Develop Python scripts to extract data from the YouTube API and store it in the MySQL database.

- Accommodate for pagination, of data given by json response.

- Implement error handling and data validation to ensure data integrity.

- Schedule periodic updates to fetch new data and keep the database up-to-date.

5. *Streamlit App Development:*

- Create a Streamlit app that provides a user-friendly interface for interacting with the YouTube channel data.

- Design the app layout with widgets for searching channels, filtering data, and displaying results.

- Implement functionality to query the MySQL database and retrieve channel details, video information, and other data as requested by the user.

- Include features for sorting, filtering, and visualizing the data to enhance user experience.

API Reference - Streamlit Docs

6. *Testing and Debugging:*

- Test the application thoroughly to identify and fix any bugs or issues.

- Perform integration testing to ensure proper interaction between the Streamlit app, Google API, and MySQL database.

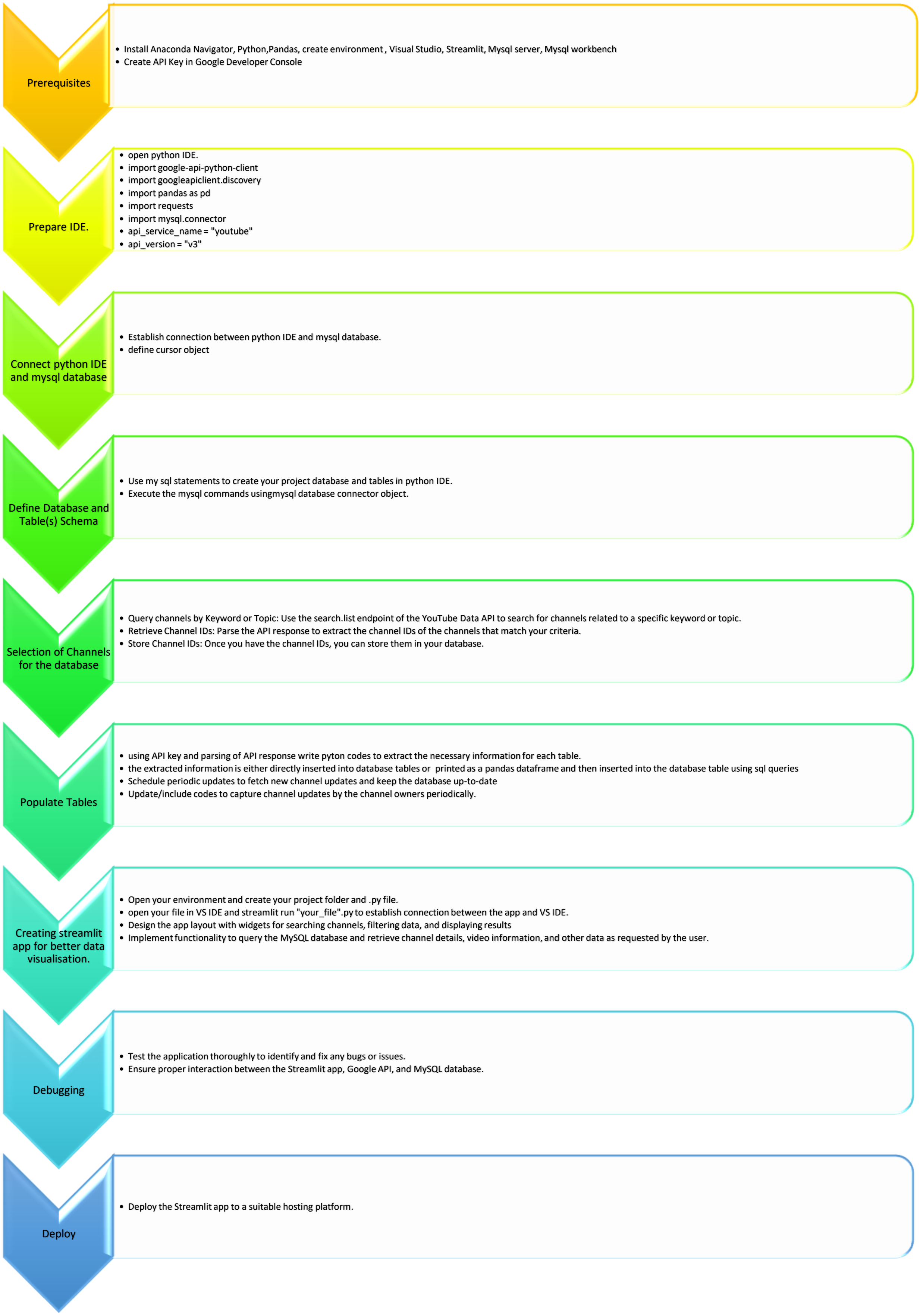
7. *Deployment:*

- Deploy the Streamlit app to a suitable hosting platform such as Heroku, AWS, or Google Cloud Platform.

- Monitor the deployed application for performance, reliability, and security.

By following these steps, you can develop a user-friendly Streamlit application that utilizes the Google API to extract information on YouTube channels, stores it in a SQL database, and enables users to search for channel details and analyze the data interactively.

WORK FLOW:



Prerequisites

- Install Anaconda Navigator, Python,Pandas, create environment , Visual Studio, Streamlit, Mysql server, Mysql workbench
- Create API Key in Google Developer Console

Prepare IDE.

- open python IDE.
- import google-api-python-client
- import googleapiclient.discovery
- import pandas as pd
- import requests
- import mysql.connector
- api_service_name = "youtube"
- api_version = "v3"

Connect python IDE and mysql database

- Establish connection between python IDE and mysql database.
- define cursor object

Define Database and Table(s) Schema

- Use my sql statements to create your project database and tables in python IDE.
- Execute the mysql commands usingmysql database connector object.

Selection of Channels for the database

- Query channels by Keyword or Topic: Use the search.list endpoint of the YouTube Data API to search for channels related to a specific keyword or topic.
- Retrieve Channel IDs: Parse the API response to extract the channel IDs of the channels that match your criteria.
- Store Channel IDs: Once you have the channel IDs, you can store them in your database.

Populate Tables

- using API key and parsing of API response write pyton codes to extract the necessary information for each table.
- the extracted information is either directly inserted into database tables or printed as a pandas dataframe and then inserted into the database table using sql queries
- Schedule periodic updates to fetch new channel updates and keep the database up-to-date
- Update/include codes to capture channel updates by the channel owners periodically.

Creating streamlit app for better data visualisation.

- Open your environment and create your project folder and .py file.
- open your file in VS IDE and streamlit run "your_file".py to establish connection between the app and VS IDE.
- Design the app layout with widgets for searching channels, filtering data, and displaying results
- Implement functionality to query the MySQL database and retrieve channel details, video information, and other data as requested by the user.

Debugging

- Test the application thoroughly to identify and fix any bugs or issues.
- Ensure proper interaction between the Streamlit app, Google API, and MySQL database.

Deploy

- Deploy the Streamlit app to a suitable hosting platform.

EXECUTION:

1. Open file “Youtube Project Code” from “Project” folder in a python IDE.
2. Insert your API Key, mysql database – host name and password, in appropriate places.
3. Run/execute cells in file starting from the first cell at top, one after the other. Read the comment given at the first line of every cell to understand what the code snippet of the particular cell accomplishes.
4. After running of all the cells, database is created, committed to memory, mysql connector object is closed and mysql database is closed.
5. Open your streamlit environment
6. Open file KNOT MY TYPE.py in a suitable visual studio IDE
7. Execute the following command in streamlit terminal to establish connection and open the stream lit app: streamlit run KNOT MY TYPE.py
8. Now explore the app to search for channel details and analyze the data.