WEEK 4

Progress Report

# Introduction

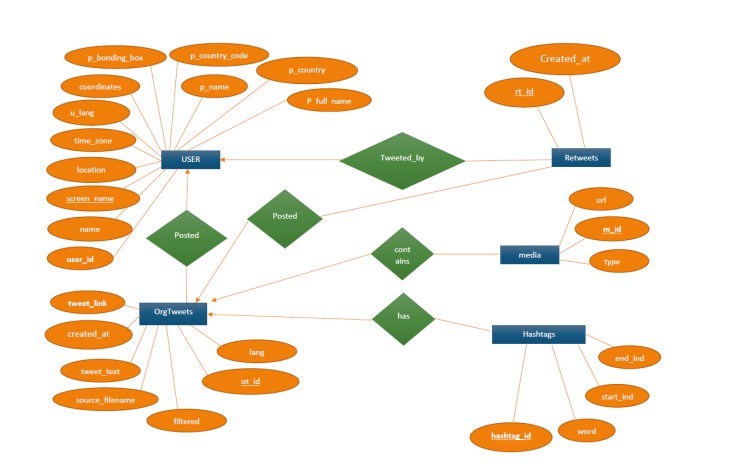
# The Twitter crawler and visualization project aims to collect data from Twitter using the Twitter API, store and organize the data in a graph database like Neo4j, and analyze and visualize the data using Python libraries. The project starts by researching the diagrams needed for the system and creating an Entity-Relationship Diagram (ERD) to model the system's data entities and their relationships. Neo4j is then selected as the database to store the collected data, given its ability to store and query graph data. Moving forward, the project seeks to refine the data collection process, improve data processing and cleaning steps, scale up the data storage solution, use more advanced techniques and tools for data analysis and visualization, integrate with other systems, and ensure proper deployment and maintenance of the system. The project aims to provide insights into Twitter users, their tweets, and their engagement levels, and enable more effective social media marketing and analysis. The Python programming language is used in the project for data collection, processing, and analysis, while Neo4j and D3.js are used for data storage and visualization, respectively.

# Prerequisites

Prerequisites for creating a Twitter crawler and visualization using Python and the Twitter API:

1. Basic knowledge of Python: You should have a basic understanding of Python programming language and syntax, including how to install and use Python libraries.
2. Twitter Developer Account: To access Twitter's API, you need to create a Twitter Developer Account, which requires a valid Twitter account.
3. Twitter API Keys: Once you have a Twitter Developer Account, you need to create a Twitter App and obtain API keys to access the Twitter API.
4. Python Libraries: You will need to install the following Python libraries: tweepy, pandas, and matplotlib. These libraries are used to access the Twitter API, handle data in dataframes, and create visualizations, respectively.
5. IDE: You should have a Python integrated development environment (IDE) installed on your computer to write and execute Python code. Some popular IDEs include PyCharm, Spyder, and Jupyter Notebook.

# ER Diagram



# Data example

Here is an example of the type of data that could be collected and analyzed in the Twitter crawler and visualization project:

{

"user\_id": 123456789,

"user\_name": "John Smith",

"user\_followers\_count": 5000,

"tweet\_id": 987654321,

"tweet\_text": "Excited to announce our new product launch! #newproduct #tech",

"tweet\_created\_at": "2023-02-22 12:30:00",

"retweet\_count": 10,

"favorite\_count": 20,

"hashtags": ["newproduct", "tech"],

"mentions": ["@company"],

"reply\_to\_tweet\_id": 123456789,

"reply\_to\_user\_id": 987654321

}

This example includes information about a Twitter user, their tweet, and engagement metrics. The data includes the user's ID, name, and number of followers, as well as the tweet's ID, text, creation time, number of retweets and favorites, hashtags, and mentions. It also includes information about whether the tweet is a reply to another tweet and, if so, the ID of the original tweet and the ID of the user who posted the original tweet.

This type of data could be used to analyze user engagement levels, track the success of a marketing campaign, identify popular hashtags and mentions, and more.

# Next week’s work

1. Data collection: Refine your data collection process to ensure that you are collecting the most relevant and useful data. This might involve using more specific search terms or filtering out irrelevant data.
2. Data processing and cleaning: Improve the data processing and cleaning steps to ensure that the data is accurate, consistent, and free of errors. This might involve using more advanced text processing techniques or incorporating machine learning algorithms to classify or cluster the data.
3. Data storage: Consider scaling up your data storage solution to accommodate larger volumes of data. You may need to migrate to a more powerful database engine or implement a distributed data storage solution.
4. Data analysis and visualization: Use more advanced techniques and tools to perform deeper analysis of the data and create more sophisticated visualizations. This might involve using machine learning algorithms to identify patterns or anomalies in the data, or creating interactive visualizations using web-based frameworks like D3.js.
5. Integration with other systems: Consider integrating your Twitter crawler and visualization system with other systems or platforms. For example, you could integrate with a web application to allow users to interact with the data in real-time, or with a marketing automation platform to provide insights into social media engagement metrics.
6. Deployment and maintenance: Ensure that your system is deployed and maintained properly to ensure ongoing stability and performance. This might involve setting up automated monitoring and alerting, or implementing automated testing and deployment pipelines.

# Conclusion:

# In conclusion, our Twitter crawler and visualization project aims to collect data from Twitter using the Twitter API, store and organize the data in a graph database like Neo4j, and analyze and visualize the data using Python libraries.

# We started by researching the diagrams needed for the system and created an Entity-Relationship Diagram (ERD) to model the system's data entities and their relationships. We then selected Neo4j as the database to store the collected data, given its ability to store and query graph data.

# Moving forward, we plan to refine our data collection process, improve our data processing and cleaning steps, consider scaling up our data storage solution, use more advanced techniques and tools for data analysis and visualization, integrate with other systems, and ensure proper deployment and maintenance of the system.

# Overall, this project aims to provide insights into Twitter users, their tweets, and their engagement levels, and enable more effective social media marketing and analysis.

# References

* Twitter API documentation: <https://developer.twitter.com/en/docs>
* py2neo documentation: <https://py2neo.org/v4/>
* Neo4j documentation: <https://neo4j.com/docs/>
* D3.js documentation: <https://d3js.org/>
* "Mining the Social Web" book by Matthew A. Russell: <http://shop.oreilly.com/product/0636920030195.do>
* "Python for Data Analysis" book by Wes McKinney: <https://www.oreilly.com/library/view/python-for-data/9781491957653/>
* "Python Data Science Handbook" by Jake VanderPlas: <https://jakevdp.github.io/PythonDataScienceHandbook/>
* "Data Science from Scratch" book by Joel Grus: <https://www.oreilly.com/library/view/data-science-from/9781492041122/>
* "Hands-On Data Science for Marketing" book by Yoon Hyup Hwang and Sinan Ozdemir: <https://www.packtpub.com/product/hands-on-data-science-for-marketing/9781800202089>