TWITTER DATA ANALYSIS:

USING PYTHON

A Project Report

Submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Kumari Shikha, Bhavana Sakat

32956, 33822

Under the esteemed guidance of

Prof. Esmita Gupta

Head of Department (IT)

DEPARTMENT OF INFORMATION TECHNOLOGY

B.K BIRLA COLLEGE OF ARTS, SCIENCE & COMMERCE

(AUTONOMOUS)

(Affiliated to University of Mumbai)

KALYAN, 421301 MAHARASHTRA

2021-2022

B. K. BIRLA COLLEGE OF ARTS, SCIENCE & COMMERCE

(AUTONOMOUS)

(Affiliated to University of Mumbai)

KALYAN – MAHARASHTRA - 421301

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled **'TWITTER DATA ANALYSIS:USING PYTHON'** is bonafied work of **BHAVANA SAKAT** AND **KUMARI SHIKHA** bearing Student ID**: 33822** and **32956 r**espectively, submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

External Examiner

Date: **College Seal**

ABSTRACT

With the social networking era rapidly evolving, social media websites are a promising platform for any individual to share their opinion, interests and ideas.

Reviews act as a valuable source of information for decision-making. Online e-commerce sites have provided their users to make their opinion about products and services. A huge amount of such opinions are publicly available in the form of reviews. Manufacturers, retailers as well as customers have great interest in customer reviews. A customer has an interest in such reviews to determine which product or a particular brand to buy. Manufacturers can analyze the improvement area in their product from such opinionated reviews. Due to a large number of reviews available on the internet for analysis, it is not cost worthy to read these manually. To optimize this time-consuming task there is a need for an automated system that provides the summarized result of user sentiments. Opinion Mining (OM) in the field of study that analyzes people's sentiments or opinions from reviews or opinionated text. In the sentiment analysis process, machine learning is used to analyze sentiments, emotions and produce a summarized result for decision making. Opinion Mining can be viewed as a natural language processing task, the task is to develop a system that understands the people's language. Opinion Mining is a difficult task due to the ambiguous nature of human languages (like English).

Twitter is one of these promising social media platforms with a gold mine of data present in it. Twitter is a platform where users exchange their ideas with something called as "tweets”. Twitter data is very much specific and almost every user's tweets are accessible to every other user or say is public. In this project, we tend to access the twitter's API and access some of the tweets to analyze them. We will try getting an access to a developer account on twitter and after that collecting some of the tweets featuring the required text using twitter's API. We will then analyze the data using Textblob and classify the tweets as positive, Negative and Neutral.
Hence, we will be able to get an idea of how a company or hashtag is received in public on a general basis.

ACKNOWLEDGEMENT

I express my sincere gratitude to the head of department **prof. Esmita gupta** and the core faculty of the **Department of technology of BK Birla college of Arts, Science and Commerce (Autonomous) Kalyan,** for providing us an opportunity to acquire knowledge from the corporate world and understand IT business better. Iam also thankful to all the professors of IT Dept for guiding and inspiring us.

I also thank the dept of IT for allowing us to work on **“Twitter Data Analysis: using Python”** project and a constant source of inspiration and guidance to us. Their valuable knowledge and experience helped us to get throughall the difficulties. I would also like to thank our friends for sharing their opinion and various inputs in a long discussion on the project.

DECLARATION

I hereby declare that the project entitled, "**Twitter Data Analysis: Using Python**" done at **B.K.Birla College**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Kumari Shikha

Bhavana Sakat

TABLE OF CONTENTS

Chapter1

Introduction

1.1 Background:-

Sentiment analysis is an excellent way to discover how people, particularly consumers, feel about a particular topic, product, or idea. The origin of sentiment analysis can be treated to the 1950s, when sentiment analysis was primarily used on written paper documents. Sentiment analysis on twitter data has been an increasing research area. Many publications published a lot of research papers on twitter data analysis.

One of those titled "A literature review on Twitter Data Analysis "by Hana Anber, Akram Salah, A.A.Abd El-Aziz. The authors have explained the growing phenomena of social media such as Facebook, LinkedIn, Twitter, and Instagram by their own characteristics and usages. Mainly they have focused on twitter for data analysis , where Twitter is an online networking service which enables users to send and read short 140-character messages called "tweets" The massive information provided by twitter such as tweet messages, user profile information, and the number of followers/ followings in the network play a significant role in data analysis. In second section authors have elaborated methods used to retrieve twitter data, twitter users rankings, and the network topology.

Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. Starting from being a document level classification, it has been handled at the sentence level and more recently at the phrase level (Wilson et al., 2005; Agarwal et al., 2009). Microblog data like Twitter, on which users post real time reactions to and opinions about "everything", poses newer and different challenges. Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009), (Bermingham and Smeaton, 2010) and Pak and Paroubek (2010). Go et al. (2009) use distant learning to acquire sentiment data. They use tweet sending in positive emotions like ":)" ":-)" as positive and negative emoticons like ":(" ":-(" as negative. They build models using Naïve Bayes, Max Ent and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space, they try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. They notethat the unigram model outperforms all other models. Specifically, bigrams and POS features do not help. Pak and Paroubek (2010) collect data following a similar distant learning paradigm.
They perform a different classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons in the same manner as Go et al. (2009). For objective data they craw] Twitter accounts of popular newspapers like "New York Times", "Washington Posts" etc. They report that POS and bigrams both help (contrary to results presented by Go et al. (2009)). Both these approaches, however, are primarily based on ngram
models. Moreover, the data they use for training and testing is collected by search queries and is therefore biased. In contrast, we present features that achieve a significant gain over a unigram baseline. In addition, we explore a different method of data representation and report significant

improvement over the unigram models. Another contribution of this paper is that we report results on manually annotated data that does not suffer from any known biases. Our data will be a random sample of streaming tweets unlike data collected by using specific queries. The size of our hand-labeled data_will allow us to perform cross validation experiments and check forth variance in performance of the classifier across folds. Another significant effort for sentiment classification on Twitter data is by Barbosa and Feng (2010). They use polarity predictions from three websites as noisy labels to train a model and use 1000 manually labeled tweets fortuning and another 1000 manually labeled tweets for testing. They however do not mention how they collect their test data. They propose the use of syntax features of tweets like retweet, hash tags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words. We extend their approach by using real valued prior polarity, and by combining prior polarity with POS. Our results show that the features that enhance the performance of our classifiers the most are features that combine prior polarity of words with their parts of speech. The tweet syntax features help but only marginally. Gamon (2004) performsentiment analysis on feedback data from Global Support Services survey. One aim of their paper is to analyze the role of linguistic features like POS tags.
They perform extensive feature analysis and feature selection and demonstrate that abstract linguistic analysis features contributes to the classifier accuracy. In this paper we perform extensive feature analysis and show that the use of only 100 abstract linguistic features performs as well as a hard unigram baseline. One fundamental problem in sentiment analysis is categorization of sentiment polarity .Given a piece of written text, the problem is to categorize the text into one specific sentiment polarity, positive or negative (or neutral). Based on the scopeof the text, there are three levels of sentiment polarity categorization, namely the document level,the sentence level, and the entity and aspect level. The document level concerns whether a document, as a whole, expresses negative or positive sentiment, while the sentence level deals with each sentence's sentiment categorization. The entity and aspect level then targets on what exactly people like or dislike from their opinions. For feature selection, Pang and Lee suggested to remove objective sentences by extracting subjective ones. They proposed a text-categorizationtechnique that is able to identify subjective content using minimum cut. Gann et al. Selected 6799 tokens based on Twitter data, where each token is assigned a sentiment score, namely TSI (Total Sentiment Index), featuring itself as a positive token or a negative token. Specifically, a TSI for a certain token is computed as: TS] = p—tp tn*np+tp tn * n where p is the number of times a token appears in positive tweets and n is the number of times a token appears in negativetweets. tp tn is the ratio of total number of positive tweets over total number of negative tweets.

Moreover, showed that using the well-known "geo-tagged" feature in Twitter to identify the polarity of political candidates in the US could be done by employing the sentiment analysis algorithms to predict the future events such as the presidential elections results. Comparing to previous approaches in sentiment topics, additional findings by showed that adding the semantic feature produces better Recall (retrieved documents) to compute the score) in negative sentimentclassification.

1.1.1 NLP

Natural Language Processing or NLP is a field of Artificial Intelligence that gives the machinesthe ability to read, understand and derive meaning from human languages.

It is a discipline that focuses on the interaction between data science and human language, and isscaling to lots of industries. Today NLP is booming thanks to the huge improvements in the access to data and the increase in computational power, which are allowing practitioners to achieve meaningful results in areas like healthcare, media, finance and human resources, among others.

1.1.1.1 USE CASES OF NLP

In simple terms, NLP represents the automatic handling of natural human language like speech or text, and although the concept itself is fascinating, the real value behind this technology comesfrom the use cases. NLP can help you with lots of tasks and the fields of application just seem to increase on a daily basis.

Let's mention some examples:

- NLP enables the recognition and prediction of diseases based on electronic health recordsand patient's own speech. This capability is being explored in health conditions that go from cardiovascular diseases to depression and even schizophrenia. For example, Amazon Comprehend Medical is a service that 2 uses NLP to extract disease conditions, medications and treatment outcomes from patient notes, clinical trial reports and other electronic health records.
- Organizations can determine what customers are saying about a service or product byidentifying and extracting information in sources like social media. This sentiment analysis can provide a lot of information about customers choices and their decision drivers.
- Companies like Yahoo and Google filter and classify your emails with NLP by analyzingtext in emails that flow through their servers and stopping spam before they even enter your inbox.
- To help identifying fake news, the NLP Group at MIT developed a new system to determine if a source is accurate or politically biased, detecting if a news source can betrusted or not.
- Amazon's Alexa and Apple's Siri are examples of intelligent voice driven interfaces thatuse NLP to respond to vocal prompts and do everything like find a particular shop, tell usthe weather forecast, suggest the best route to the office or turn on the lights at home.

NLP is particularly booming in the healthcare industry. This technology is improving care delivery, disease diagnosis and bringing costs down while healthcare organizations are going through a growing adoption of electronic health records. The fact that clinical documentation canbe improved means that patients can be better understood and benefited through better healthcare. The goal should be to optimize their experience, and several organizations are already working on this. Basically, they allow developers and businesses to create a software thatunderstands human language. Due to the complicated nature of human language, NLP can be difficult to learn and implement correctly. However, with the knowledge gained from this article,you will be better equipped to use NLP successfully, no matter your use case.

Natural language processing has a wide range of applications in business. As just one example,brand sentiment analysis is one of the top use cases for NLP in business. Many brands track sentiment on social media and perform social media sentiment analysis.

In social media sentiment analysis, brands track conversations online to understand what customers are saying, and glean insight into user behavior. "One of the most compelling ways NLP offers valuable intelligence is by tracking sentiment — the tone of a written message (tweet, Facebook update, etc.) — and tag that text as positive, negative or neutral," says Rehling.Similarly, Facebook uses NLP to track trending topics and popular hashtags.

1.2) Objectives:

In our project we will be analyzing the mined data from Twitter and can get a general knowledgeof what is the public opinion on certain issues, hash tags etc. Our project will also be helpful to any organization that wants to know how they are being received in general public.

•Running analysis on specific users, and the way they move with the world.

•Running analysis on specific keyword and visualizing the sentiment.

1.3) Purpose, Scope and Applicability:

1.3.1) Purpose:

Imagine you just launched a new product feature and notice a sharp increase in mentions onTwitter.

Are customers tweeting more because they are delighted with the new feature? Or, are theyactually complaining about the feature?

Going through each of these comments manually would take far too much time.

You did miss out on valuable feedback that could help you instantly improve a customer'sexperience with the latest feature (bug issues, user experience).

By performing analysis with NLP, you can quickly understand the tone and context of socialmentions on Twitter.

Sentiment analysis refers to identifying as well as classifying the sentiments that are expressed inthe text source. Tweets are often useful in generating a vast amount of sentiment data upon analysis. These data are useful in understanding the opinion of the people about a variety of topics.

Twitter sentiment analysis will allow us to keep track of what’s being said about a product orservice or topic on social media, and can help detect negative mentions before they escalate.

1.3.2) Scope:

For an organization, big data analytics can provide insights that surpass human capability. Being able to run large amounts of data through computation-heavy analysis is something mathematicalmodels and machines thrive at. Mining focuses on the discovery of meaningful knowledge from data such as online mailing lists, blogs, and social media and includes analysis of structure, usageand content. Web content mining aims to extract and analyze useful information (e.g., opinions, sentiment, main topics) from web content by applying techniques from multidisciplinary fields including data mining, machine learning, natural-language processing, information retrieval, and statistics. Following the traditional framework of general data mining, a typical content-mining process includes preparing data so they can be imported and read in data-mining software, reducing the dimensionality of data, applying classic data-mining techniques, and terminating or iterating the process according to interpretation.

This project will be helpful to the companies , political parties as well as to the common people.It will be helpful to political party for reviewing about the program that they are going to do or the program that they have performed. Similarly companies also can get review about their newproduct on newly released hardware or software. Also the movie maker can take review on howtheir movie is being received by the public through sentiment analysis of negative, positive or neutral.

1.3.3) Applicability:

Analyze data: (1) calculate frequency vectors to create term-Tweet frequency table resulting in identification of frequently occurring terms (e.g., marijuana smoking, sugar, milk, and genetics);

(2) compare terms across Tweet corpuses using chi-square tests (e.g., BMI corpus distinct terms include measure, circumference, supplements, height, and calculator, and obesity corpus distinct terms include family, marijuana, smoking, and milk;

(3) apply sentiment analysis to identify topics associated with positive (e.g., obesity, BMI) ornegative (e.g., body fat, appetite)

Chapter 2

Survey of Technologies

2.1) Twitter:

Twitter as Big Data, is exactly what it sounds like —a lot of data. Alone, a single pointof data can't give you much insight. But terabytes of data, combined together with complex mathematical models and boisterous computing power, can create insights human beings aren't capable of producing.
Twitter is a gold mine of data. Unlike other social platforms, almost every user's tweets are completely public and pullable. This is a huge plus if you're trying to get alarge amount of data to run analytics on. Twitter data is also pretty specific. Twitter'sAPI allows you to do complex queries like pulling every tweet about a certain topic within the last twenty minutes, or pull a certain user's non-retweeted tweets.
A simple application of this could be analyzing how your company is received in the general public. You could collect the last 2,000 tweets that mention your company (orany term you like), and run a sentiment analysis algorithm over it.

2.2) Twitter API:

The Twitter API is a set of programmatic endpoints that can be used to understand or build the conversation on Twitter. This API allows you to find and retrieve, engage with, or create a variety of different resources including the following: Tweets. Twitter API platform provides broad access to public Twitter data that users have chosen to share withthe world. It also support APIs that allow users to manage their own non-public Twitter information (e.g., Direct Messages) and provide this information to developers whom they have authorized to do so.

2.2.1) How to get access to the Twitter API: -

Step one	: Sign up for a developer account
Step two	: Save your App's key and tokens and keep them secure.
Step three	**:** Make your first request
[Optional] Step four	**:** Apply for additional access.

2.3) Tweepy:

Tweepy is an open-sourced, easy-to-use Python library for accessing the Twitter API. Itgives you an interface to access the API from your Python application. Tweepy includesa set of classes and methods that represent Twitter's models and API endpoints, and it transparently handles various implementation details, such as: Data encoding and decoding.

2.4) Python:

Python is an interpreter, object-oriented, high-level programming language with dynamicsemantics.

Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together all the main logic coding has been implemented in Python.

Why Python?

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encouragesprogram modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, andcan be freely distributed.

2.5) PyCharm:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. PyCharm provides smart code completion, code inspections, on-the- fly error highlighting and quick-fixes, along with automated code refactoring and rich navigation capabilities. PyCharm's huge collection of tools out of the box includes an integrated debugger and test runner; Python profiler; a built-in terminal; integration with major VCS and built-in database tools; remote development capabilities with remote interpreters; an integrated terminal; and integration with Docker and Vagrant. We can usePyCharm on Windows, macOS and Linux with a single license key.

2.6) Streamlit:

Streamlit is an open source app framework in Python language. It helps us create webapps for data science and machine learning in a short time. It is compatible with majorPython libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.

Streamlit allows you to write an app the same way you write a python code. Streamlit makes it seamless to work on the interactive loop of coding and viewing results in the web app. The streamlit has a distinctive data flow, any time something changes in your code or anything needs to be updated on the screen, streamlit reruns your python script entirely from the top to the bottom. This happens when the user interacts with the widgetslike a select box or drop-down box or when the source code is changed.

- **How to install streamlit in PyCharm using terminal**
  : "pip install streamlit"

2.7) NLP:

Natural language processing (NLP) is the ability of a computer program to understandhuman language as it is spoken and written -- referred to as natural language. It is a component of artificial intelligence (AI).

Sentiment Analysis (also known as opinion mining or emotion AI) is a sub-field of NLP that tries to identify and extract opinions within a given text across blogs, reviews,social media, forums, news etc.

2.8) Python Libraries\ Packages:

2.8.1) TextBlob

TextBlob is a Lexicon-based sentiment analyzer It has some predefined rules or we can say word and weight dictionary, where it has some scores that help to calculate a sentence's polarity. That's why the Lexicon-basedsentiment analyzers are also called “Rule-based sentiment analyzers”. TextBlob has semantic labels that help with fine-grained analysis. For example — **emoticons, exclamation mark, emojis**, etc. Subjectivity lies between [0,1]. Subjectivity quantifies the amount of personal opinion and factual information contained in the text.

TextBlob is useful for Twitter Sentiment Analysis Python in the following ways:

Tokenization: TextBlob can tokenize the text blocks into different sentences and words.

This makes reading between the lines much easier.

Noun Phrases Extraction using TextBlob:

The noun is mostly used as an Entity in sentences. It is also one the most important NLP utility in Dependency Parsing. This is how different nouns are extracted from a sentence using TextBlob –

Part-of-Speech Tagging using TextBlob:

TextBlob is also used for tagging parts of speech with your sentences.

2.8.2) Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. Pandas allow us to analyze big data andmake conclusions based on statistical theories. Pandas can clean messy data sets,and make them readable and relevant. Relevant data is very important in data science. A pandas Series is a **one-dimensional labelled data structure which can hold data such as strings, integers and even other Python objects**. It is built on top of numpy array and is the primary data structure to hold one-dimensional data in pandas. In Python, a pandas Series can be created using the constructor pandas.

2.8.3) Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python. NumPy aims to provide an array object that is up to 50 xs faster than traditional Python lists.

2.8.4) Seaborn

Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

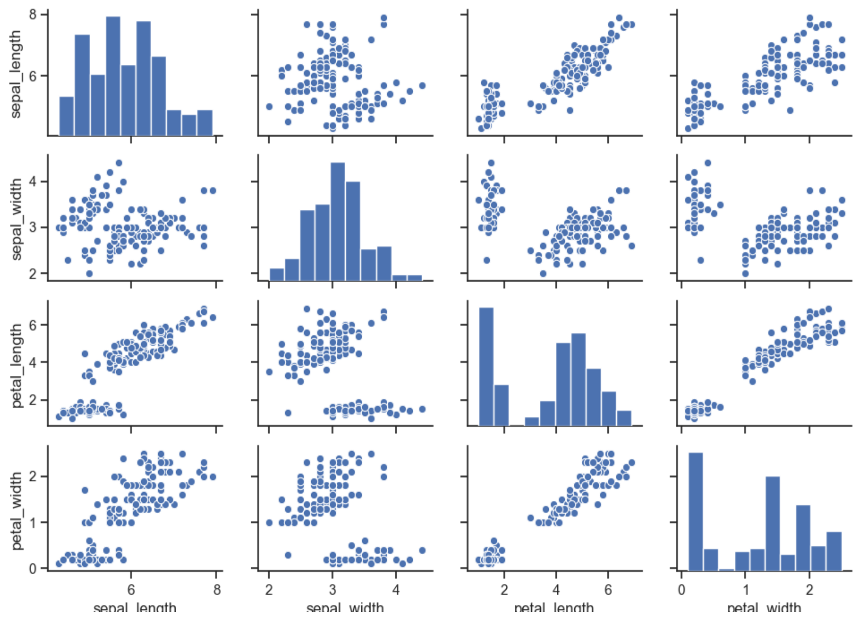


Fig 2.1: Seaborn pairplot example

2.8.5) Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications. Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

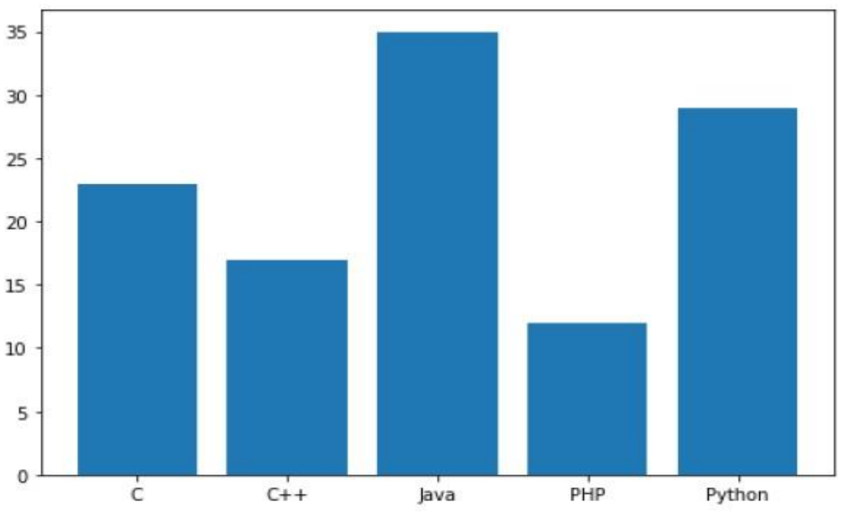


Fig 2.2: Matplotlib - Bar Plot

2.8.6) PIL

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

2.8.7) SciPy

SciPy in Python is an open-source library used for solving mathematical, scientific, engineering, and technical problems. It allows users to manipulate the data and visualize the data using a wide range of high-level Python commands. SciPy is built on the Python NumPy extention. SciPy is also pronounced as “Sigh Pi.”

- NumPy vs SciPy

Both NumPy and SciPy are Python libraries used for used mathematical and numerical analysis. NumPy contains array data and basic operations such as sorting, indexing, etc whereas, SciPy consists of all the numerical code. Though NumPy provides a number of functions that can help resolve linear algebra, Fourier transforms, etc, SciPy is the library that actually contains fully-featured versions of these functions along with many others. However, if you are doing scientific analysis using Python, you will need to install both

NumPy and SciPy since SciPy builds on NumPy.

2.8.8) Pyzmq

PyZMQ is the Python bindings for ØMQ. This documentation currently contains notes on some important aspects of developing PyZMQ and an overview of what the ØMQ API looks like in Python.

- Zmq: - ZeroMQ is a library used to implement messaging and communication systems between applications and processes - fast and asynchronously.

Chapter 3

Requirements and Analysis

3.1) Problem Definition:

As the social networking era is rapidly evolving, millions of people are on internet via social media, mainly focusing on twitter.
Twitter users often tend to share their opinion on certain issues, hashtags, product or service etc.
Our project will tend to focus on helping a company or organizations or even an individual to get to knowhow their product or in the general public in the form of positive, negative or neutral remark.

3.2) Requirement specification:

3.2.1) Python 3.10:

Python 3.10 adds - **structural pattern matching, parenthesized contextmanagers, more typing, and the new and improved error messages**.
It has a simple syntax that mimics natural language, so it's easier to read and understand. This makes it quicker to build projects, and faster to improve on them. It's versatile. Python can be used for many different tasks, from web development, software development, data science andmachine learning. Python is beginner friendly and easy to learn. It's open source. Python has a large and active community that contributes to Python's pool of modules and libraries, and actsas a helpful resource for other programmers.

3.2.2) PyCharm :

PyCharm is an integrated development environment used in computer programming, specifically for the Python programming language. It is developed by the Czechcompany JetBrains.

3.3) Planning and Scheduling:

3.3.1) Gantt Chart:

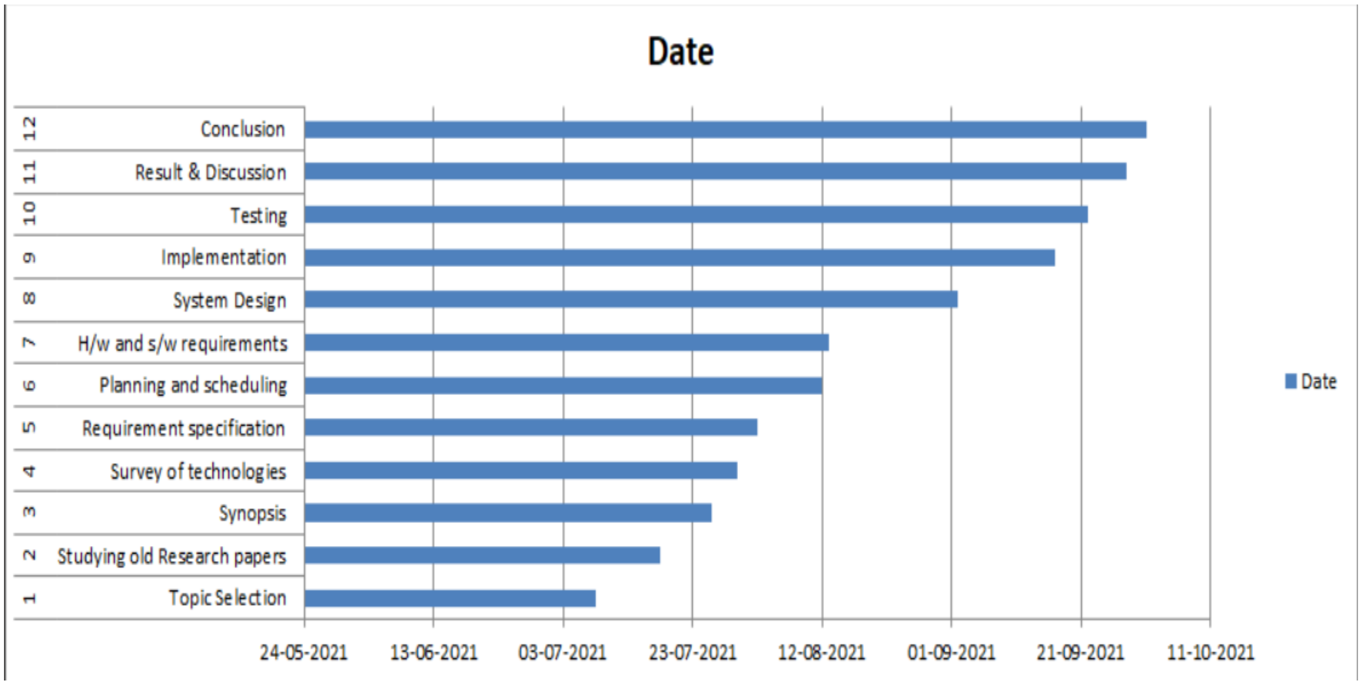
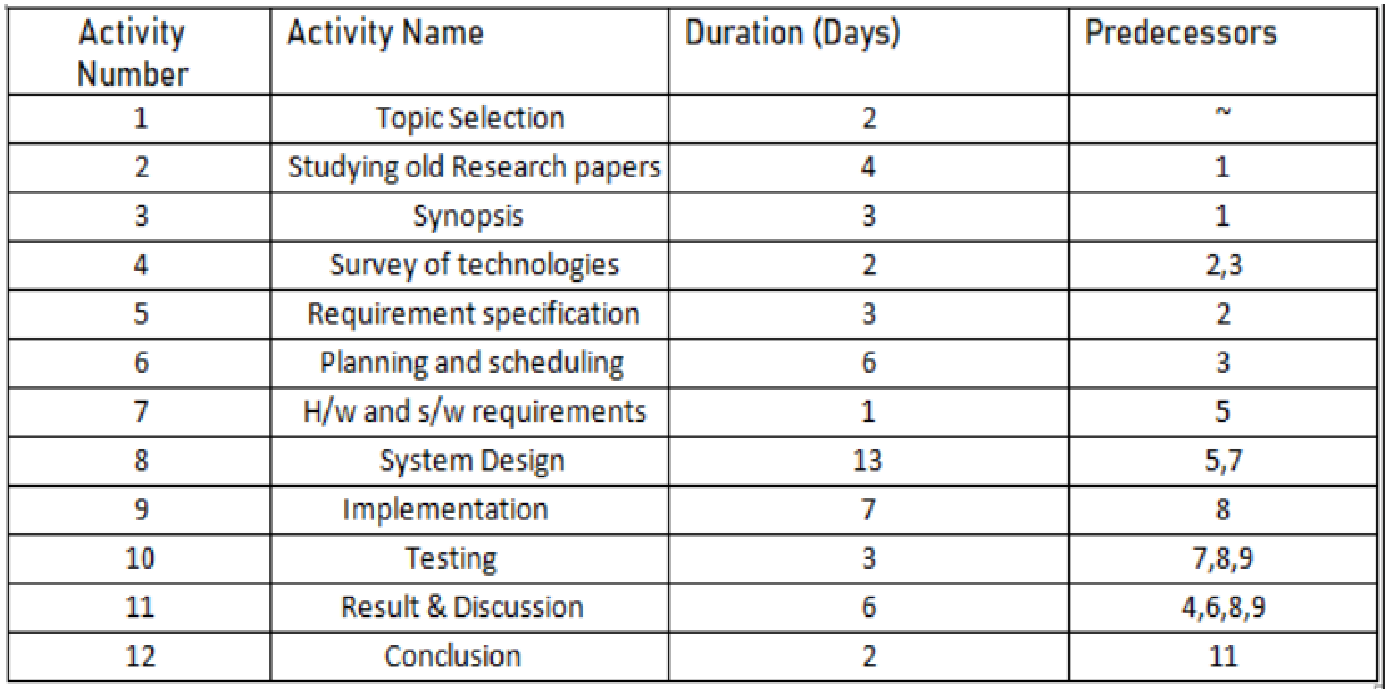


Fig 3.1: Gantt Chart

3.3.2) PERT Chart:



Activity Number	Activity Name	Duration (Days)	Predecessors
1	Topic Selection	2	~
2	Studying old Research papers	4	1
3	Synopsis	3	1
4	Survey of technologies	2	2,3
5	Requirement specification	3	2
6	Planning and scheduling	6	3
7	H/w and s/w requirements	1	5
8	System Design	13	5,7
9	Implementation	7	8
10	Testing	3	7,8,9
11	Result & Discussion	6	4,6,8,9
12	Conclusion	2	11

Fig 3.2: PERT Chart

3.4) Hardware and Software Requirements:

3.4.1) Software

Operating System: - Windows 10 / 11

PyCharm Community Edition 2021.3.2

Microsoft World 2010

Notepad++

3.4.2) Hardware Specifications:

1. Processor	: Intel i5
2. Motherboard	: Intel ®Chipset Motherboard
3. RAM	: 8GB
4. Hard disk	: 1 TB
5. Monitor	: 1024×720 display
6. Speed	: 2.7GHZ

3.5) Conceptual Models:

3.5.1) SYSTEM FLOWCHART:

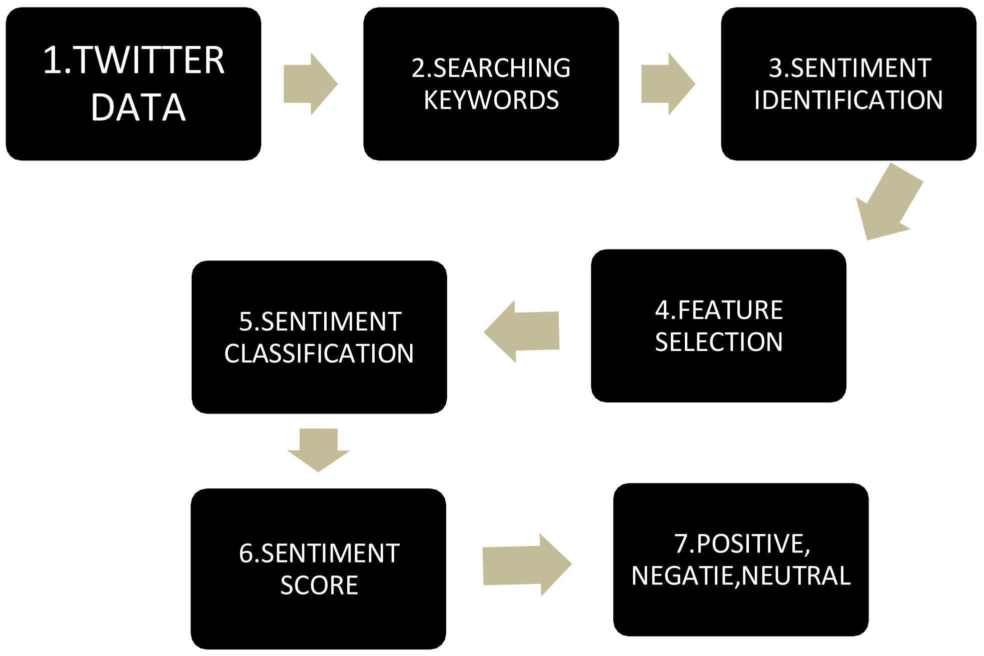

1.TWITTER DATA
2.SEARCHING KEYWORDS
3.SENTIMENT IDENTIFICATION
4.FEATURE SELECTION
5.SENTIMENT CLASSIFICATION
6.SENTIMENT SCORE
7.POSITIVE, NEGATIE,NEUTRAL

Fig 3.3 : System Flowchart

3.5.2) DFD:

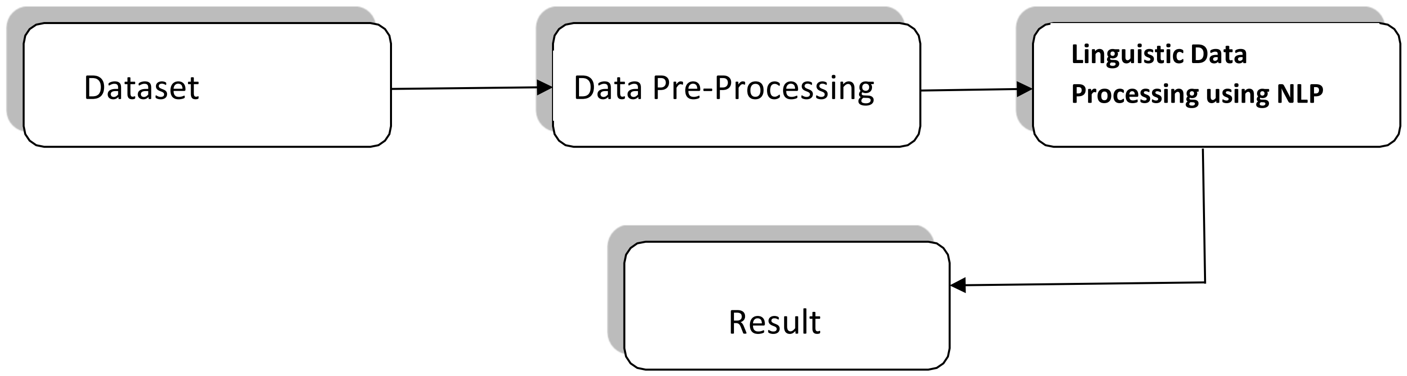


Fig 3.4: Data Flow Diagram

3.6) Architecture:

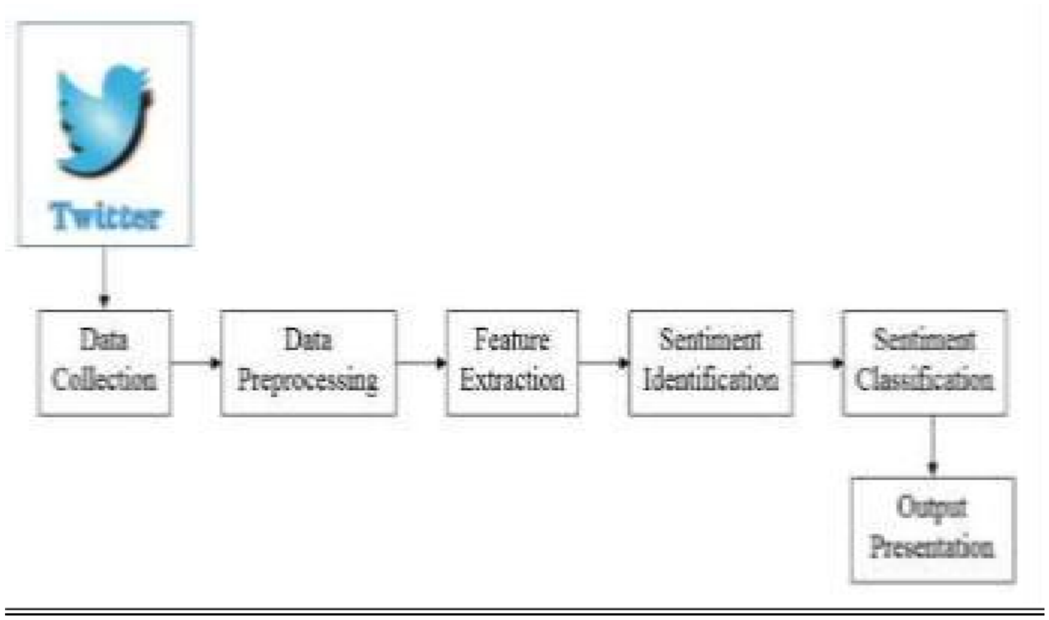


Fig.3.5: Basic Architecture of Proposed Work

3.6.1) Data Collection Process :

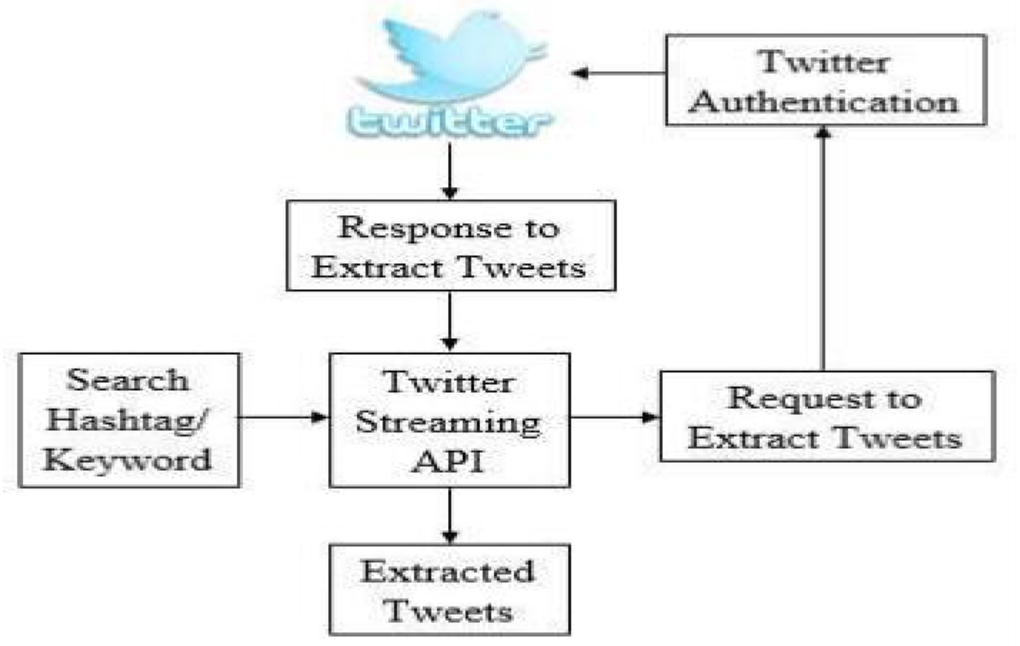
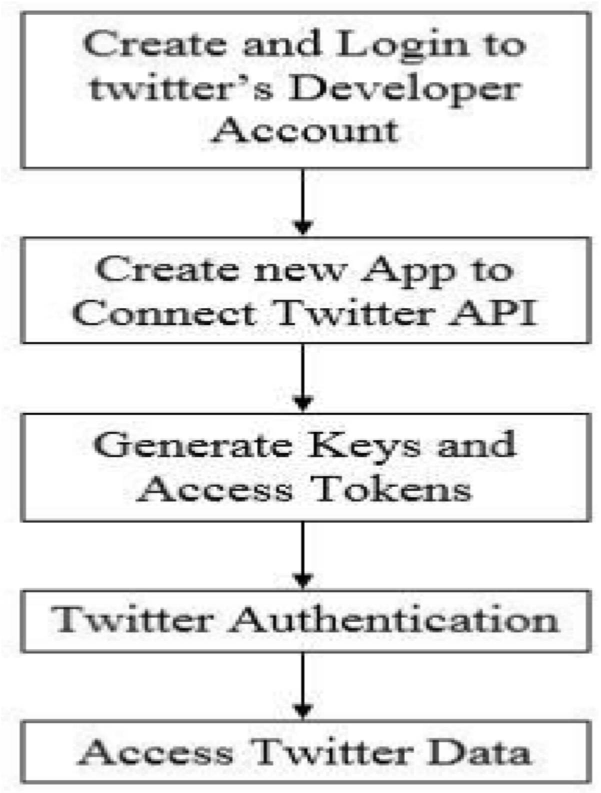


Fig. 3.6: Data Collection Process

3.6.2) Data Authentication Process:



Create and Login to twitter's Developer Account
Create new App to Connect Twitter API
Generate Keys and Access Tokens
Twitter Authentication
Access Twitter Data

Fig. 3.7: Twitter Authentication

3.7) User Characteristics :

Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech , tweets and database sources through Natural Languageprocessing(NLP).

Sentiment analysis involves classifying opinions in text into categories like “positive”, “negative” or “neutral”. It’s also referred as subjectivity analysis, opinion mining and appraisalextraction.

The words opinion, sentiment, view and belief are used interchangeably but there are differencesbetween them.

- Opinion: A conclusion open to dispute (because different experts have different opinions)View:
- Subjective opinion
- Belief: Deliberate acceptance and intellectual assentSentiment: Opinion representing one’s feelings.
-

Sentiment analysis is a term that include many tasks such as sentiment extraction, sentiment classification, subjectivity classification, summarization of opinions or opinion spam detectionamong others.

It aims to analyze people’s sentiments, attitudes, opinions emotions, etc. towards elements suchas products, individuals, topics, organization and services.

3.8) FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

3.8.1) TECHNICAL FEASIBILITY:

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployedand highly extensible.

3.8.2) ECONOMIC FEASIBILITY:

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn’t incur any kind of drastic increase in the expenses. Python is open source and ready available for all users. Since the project is runned in python and pycharm hence is cost efficient.

3.9) SOFTWARE SPECIFICATION

3.9.1) NLP Analysis

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English.
Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinicalexpert system, etc.
The field of NLP involves making computers to perform useful tasks with the natural Languages humans use. The input and output of an NLP system can be –

- Speech
- Written text

Their application to Natural Language Processing (NLP) was less impressive at first, but has now proven to make significant contributions, yielding state-of-the-art results for some commonNLP tasks. Named entity recognition (NER), part of speech (POS) tagging or sentiment analysisare some of the problems where neural network models have outperformed traditional approaches. The progress in machine translation is perhaps the most remarkable among all.
NLP is a tool for computers to analyze, comprehend, and derive meaning from natural language in an intelligent and useful way. This goes way beyond the most recently developed chatbots andsmart virtual assistants. In fact, natural language processing algorithms are everywhere from search, online translation, spam filters and spell checking.

3.9.2) API:

API is an abbreviation for Application Programming Interface which is a collection of communication protocols and subroutines used by various programs to communicate between them. A programmer can make use of various API tools to make its program easier and simpler.Also, an API facilitates the programmers with an efficient way to develop their software programs. Thus in simpler terms, an API helps two programs or applications to communicate with each other by providing them with necessary tools and functions. It takes the request from the user and sends it to the service provider and then again sends the result generated from the service provider to the desired user.

A developer extensively uses API's in his software to implement various features by using an API call without writing the complex codes for the same. We can create an API for an operatingsystem, database systems, hardware system, for a JavaScript file or similar object oriented files.Also, an API is similar to a GUI(Graphical User Interface) with one major difference. Unlike GUI's, an API helps the software developers to access the web tools while a GUI helps to makea program easier to understand by the users.

Real life example of an API:

Suppose, we are searching for a hotel room on an online website. In this case, you have a vast number of options to choose from and this may include the hotel location, the check-

in andcheck-out dates, price, accommodation details and many more factors. So in order to
book the

room online, you need to interact with the hotel booking's website which in further will let you know if there is a room available on that particular date or not and at what price. Now in the above example, the API is the interface that actually communicates in between. It takes the request of the user to the hotel booking's website and in turn returns back the most relevant datafrom the website to the intended user. Thus, we can see from this example how an API works and it has numerous applications in real life from switching on mobile phones to maintaining a large amount of databases from any corner of the world.
There are various kinds of API's available according to their uses and applications like the Browser API which is created for the web browsers to abstract and to return the data from surroundings or the Third party API's, for which we have to get the codes from other sites on theweb(e.g. Facebook, Twitter).

ADVANTAGES OF APIS

 Efficiency: API produces efficient, quicker and more reliable results than the outputsproduced by human beings in an organization.

 Flexible delivery of services: API provides fast and flexible delivery of services according todeveloper's requirements.

 Integration: The best feature of API is that it allows movement of data between various sitesand thus enhances integrated user experience.

 Automation: As API makes use of robotic computers rather than humans, it produces betterand automated results.

 New functionality: While using API the developers find new tools and functionality for APIexchanges.

DISADVANTAGES OF APIS

 Cost: Developing and implementing API is costly at times and requires high maintenance andsupport from developers.

 Security issues: Using API adds another layer of surface which is then prone to attacks, andhence the security risk problem is common in API's.

END POINTS

What is an API Endpoint?

Simply put, an endpoint is one end of a communication channel. When an API interacts with another system, the touchpoints of this communication are considered endpoints. For APIs, anendpoint can include a URL of a server or service. Each endpoint is the location from which APIs can access the resources they need to carry out their function.
APIs work using 'requests' and 'responses.' When an API requests information from a web application or web server, it will receive a response. The place that APIs send requests and wherethe resource lives, is called an endpoint.

Why Are API Endpoints Important?

All over the world, companies leverage APIs to transfer vital information, processes, transactions, and more. API usage will only increase as time goes on, and making sure that each touchpoint in API communication is intact is vital to the success of each API. Endpoints specify where resources can be accessed by APIs and play a key role in guaranteeing the correct functioning of the software that interacts with it. In short, API performance relies on its ability tocommunicate effectively with API Endpoints.

Do I Need to Monitor API Endpoints?

YES. Understanding how each API is performing can drastically change the way you're able to capture the value APIs add to your business. Proactively Monitoring APIs can ensure that you'reable to find issues before real users experience them.

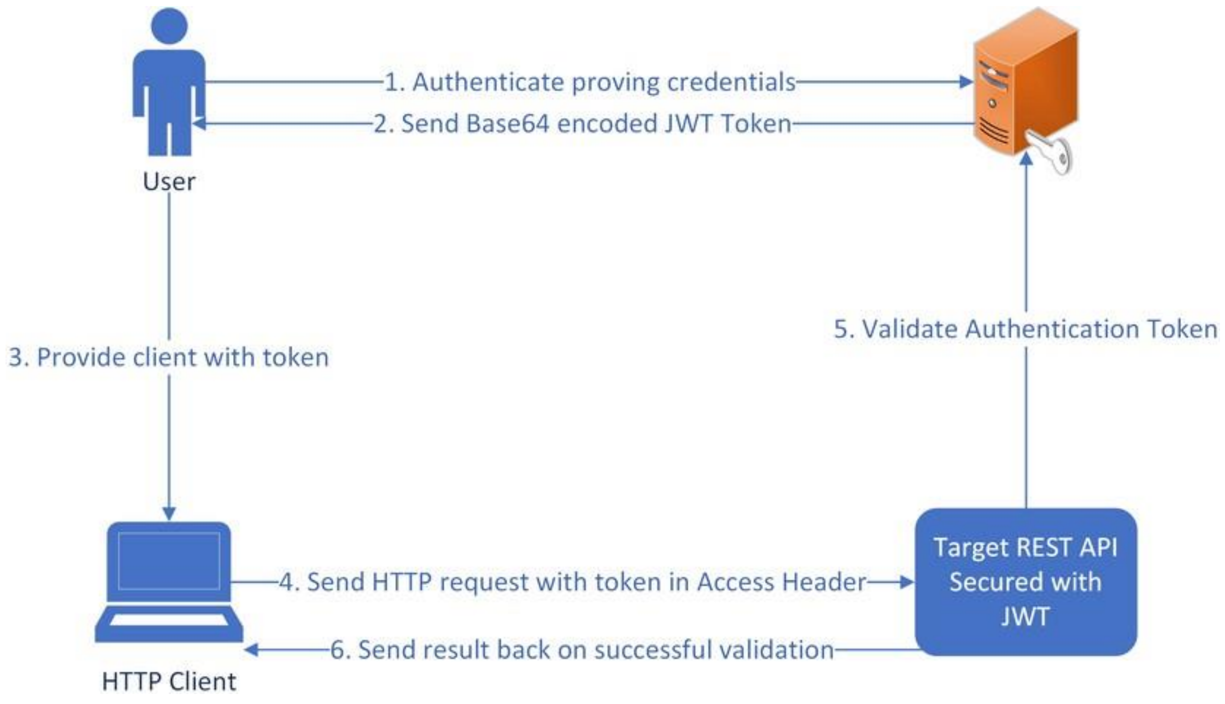

User
1. Authenticate proving credentials
2. Send Base64 encoded JWT Token
3. Provide client with token
5. Validate Authentication Token
4. Send HTTP request with token in Access Header
Target REST API Secured with JWT
6. Send result back on successful validation
HTTP Client

Fig 3.8: REST API

REST stands for REpresentational State Transfer and API stands for Application Program Interface. REST is a software architectural style that defines the set of rules to be used for creating web services. Web services which follow the REST architectural style are known as RESTful web services. It allows requesting systems to access and manipulate web resources byusing a uniform and predefined set of rules. Interaction in REST based systems happen throughInternet's Hypertext Transfer Protocol (HTTP).

A Restful system consists of a:

- client who requests for the resources.server who has the resources.
-

It is important to create REST API according to industry standards which results in ease ofdevelopment and increase client adoption.

3.9.3) VISUALIZATION

Data visualization is a way of exploring **complex patterns** or **large quantities** of data that cannot be easily perceived by looking at a table of numbers or reading paragraphs of text. Thegoal of data visualization is to communicate information more clearly, and it does so by employing our innate ability to recognize visual patterns in our environment.

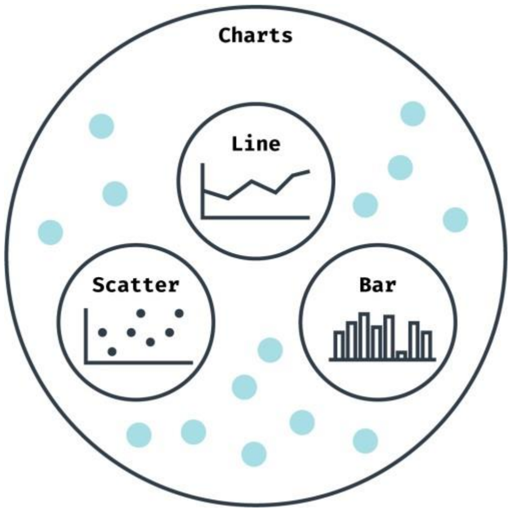


Fig 3.9 Visualization Diagram

Some data visualizations are **exploratory** in that they are created before any analysis is done onthe data. Looking at a visual representation of our dataset can give us clues about what to focuson during analysis.

Some data visualizations are **communicative** in that they are created in order to present our analysis findings to an audience. Using visual patterns to represent patterns in data can be aneffective way of explaining complex results.
Ultimately, data visualizations can more effectively answer questions, tell stories and put fortharguments than words alone.

In the world of Big Data, data visualization tools and technologies are essential to analyzemassive amounts of information and make data-driven decisions.

Our eyes are drawn to colours and patterns. We can quickly identify red from blue, square from circle. Our culture is visual, including everything from art and advertisements to TV and movies.Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we quickly see trends and outliers. If we can see something, we internalize it quickly. It's storytelling with a purpose. If you've ever stared at a massive spreadsheet of data and couldn't see a trend, you know how much more effective a visualization can be.

Big Data is here and we need to know what it says

As the "age of Big Data" kicks into high-gear, visualization is an increasingly key tool to makesense of the trillions of rows of data generated every day. Data visualization helps to tell storiesby curating data into a form easier to understand,
highlighting the trends and outliers. A good visualization tells a story, removing the noise fromdata and highlighting the useful information.
However, it's not simply as easy as just dressing up a graph to make it look better or slapping onthe "info" part of an infographic. Effective data visualization is a delicate balancing act betweenform and function. The plainest graph could be too boring to catch any notice or it make tell a powerful point; the most stunning visualization could utterly fail at conveying the right messageor it could speak volumes. The data and the visuals need to work together, and there's an art to combining great analysis with great storytelling.
Why data visualization is important for any career
It's hard to think of a professional industry that doesn't benefit from making data moreunderstandable. Every STEM field benefits from understanding data—and so
do fields in government, finance, marketing, history, consumer goods, service industries,education, sports, and so on.
While we'll always wax poetically about data visualization (you're on the Tableau website, afterall) there are practical, real-life applications that are undeniable. And, since visualization is so prolific, it's also one of the most useful professional skills to develop. The better you can conveyyour points visually, whether in a dashboard or a slide deck, the better you can leverage that information.

CoderzColumn
Data Visualization - Basic Principals Of Information Visualization

Fig 3.10: Types Of Visualizations

The concept of the citizen data scientist is on the rise. Skill sets are changing to accommodate a data-driven world. It is increasingly valuable for professionals to be able to use data to make decisions and use visuals to tell stories of when data informs the who, what, when, where, and how. While traditional education typically draws a distinct line between creative storytelling andtechnical analysis, the modern professional world also values those who can cross between the two: data visualization sits right in the middle of analysis and visual storytelling.

Chapter 4
System Design

4.1. Basic Modules:-

- Twitter Data Collection
- Data Preprocessing
- Text Blob

4.1.1) TWITTER DATA COLLECTION

Twitter is a gold mine of data. Unlike other social platforms, almost every user's tweets are completely public and pullable. This is a huge plus if you're trying to get a large amount of data to run analytics on. Twitter data is also pretty specific.

Twitter's API allows you to do complex queries like pulling every tweet about a certain topic within the last twenty minutes, or pull a certain user's non retweeted tweets. A simple application of this could be analyzing how your company is received in the general public. You could collect the last 2,000 tweets that mention your company (or any term you like), and run a sentiment analysis algorithm over it.25

Twitter's Developer Policy is generally interpreted as allowing sharing of tweets locally, i.e., within an academic institution. For example, we share the datasets we have collected at GW Libraries with members of the GW research community (but when sharing outside the GW community, we only sharethe tweet ids).

However, only a small number of institutions proactively collect Twitter data – your library is a goodplace to inquire. Another option for acquiring an existing Twitter dataset is Tweet Sets, a web
application that I've developed. Tweet Sets allows you to create your own dataset by querying and limiting an existing dataset. For example, you can create a dataset that only contains original tweets with the term "trump" from the Women's March dataset. If you are local, Tweet Sets will allow you to
download the complete tweet; otherwise, just the tweet ids can be downloaded. Currently, Tweet Setsincludes nearly a half billion tweets.

We can also target users that specifically live in a certain location, which is known as spatial data. Another application of this could be to map the areas on the globe where your company has been mentioned the most. As you can see, Twitter data can be a large door into the insights of the general public, and how they receive a topic. That, combined with the openness and the generous rate limitingof Twitter's API, can produce powerful results.

4.1.2) DATA PREPROCESSING

Data preprocessing is an important tool for Data Mining (DM) algorithm. Twitter data is an unstructureddata set it is a collection of information from people entered his/her feelings, opinion, attitudes, products review, emotions, etc. This type of information is growing day by day in the internet. May companies want to analyze customers opinions which like the product and the services. The Proposed work to analyses the twitter trending information and collect various different information form the users. It improves the accuracy of Twitter data. This work easy to identify the people reaction or opinion. Additionally, improve the better performance for data preprocessing tool.

4.1.3) TEXTBLOB

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving intocommon natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Install TextBlob using the following commands in terminal: pip install -U textblob This will install TextBlob and download the necessary NLTK corpora. The above installation will take quite some timedue to the massive amount of tokenizers, chunkers, other algorithms, and all of the corpora to be downloaded. Some terms that will be frequently used are :26

- Corpus – Body of text, singular. Corpora is the plural of this.
- Lexicon – Words and their meanings.
- Token – Each “entity” that is a part of whatever was split up based on rules.

For examples, each word is a token when a sentence is “tokenized” into words. Each sentence can also be a token

So basically tokenizing involves splitting sentences and words from the body of the text. The approachthat the TextBlob package applies to sentiment analysis differs in that it’s rule-based and therefore requires a pre-defined set of categorized words. These words can, for example, be uploaded from theNLTK database. Moreover, sentiments are defined based on semantic relations and the frequency of each word in an input sentence that allows getting a more precise output as a result.

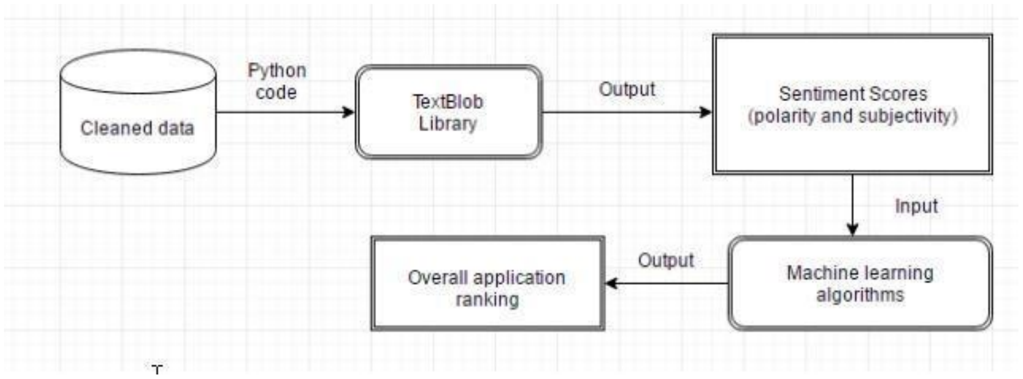
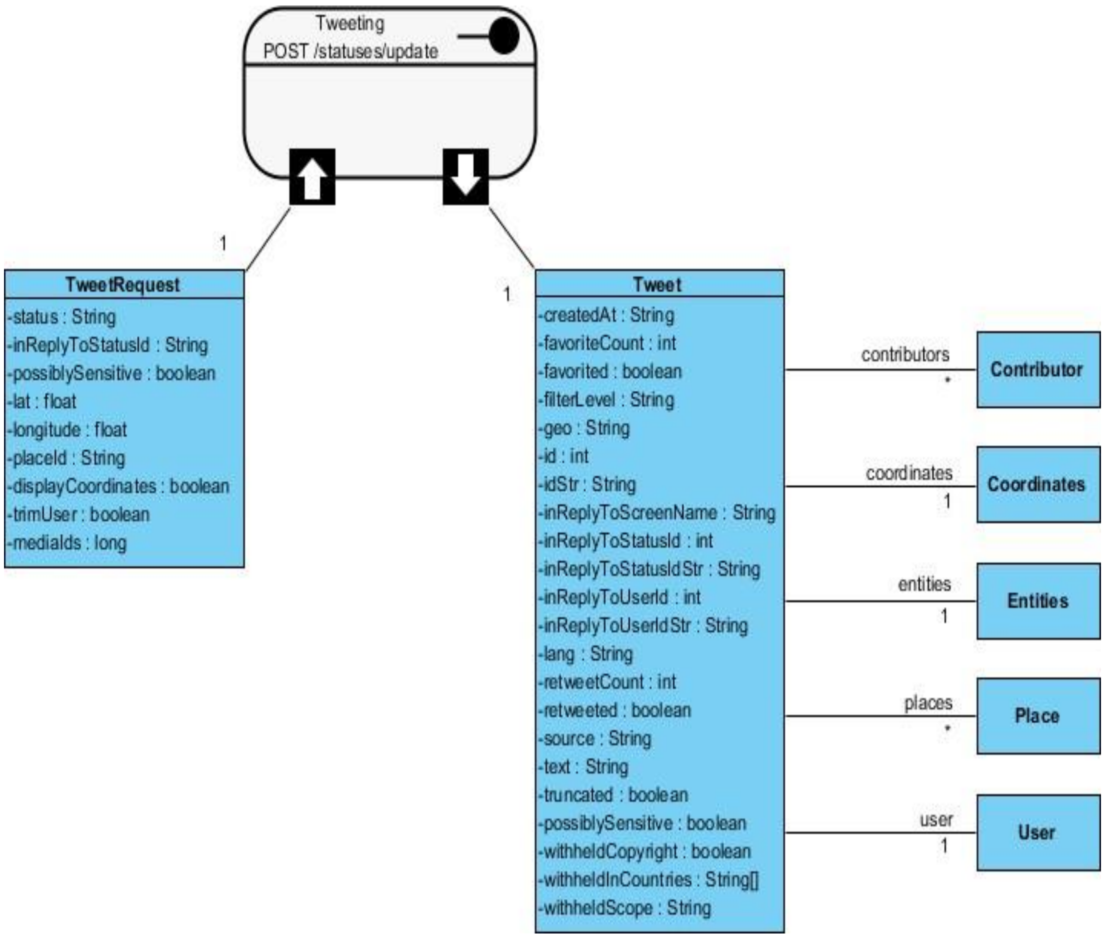

Cleaned data
Python code
TextBlob Library
Output
Sentiment Scores (polarity and subjectivity)
Input
Machine learning algorithms
Output
Overall application ranking

Fig 4.1: TextBlob Architecture Diagram

TextBlob's output for a **polarity** task is a float within the range [-1.0, 1.0] where -1.0 is a negative polarity and 1.0 is positive. This score can also be equal to 0, which stands for a neutralevaluation of a statement as it doesn't contain any words from the training set.

Whereas, a **subjectivity/objectivity** identification task reports a float within the range [0.0, 1.0]where 0.0 is a very objective sentence and 1.0 is very subjective.

4.2) Schema Design:



4.3) Data Integrity & Constraints:

1. Downloading the data from Twitter using tweepy API

2. We will now analyze the sentiments of tweets that we have downloaded and then visualizethem. (Sentiment Analysis)

3. We will create a data frame of all the tweet data that we have downloaded. Later all theprocessed data will be saved to a CSV file in the local system.

4. Cleaning Tweet Texts using NLP Operations

5. After performing the NLP operations, we visualize the most frequent words in the tweetsthrough a word cloud and using the term frequency.

6. Analyzing Highest Occurring Words Term Frequency.

Chapter 5
Implementation and testing

5.1) Implementation approaches :

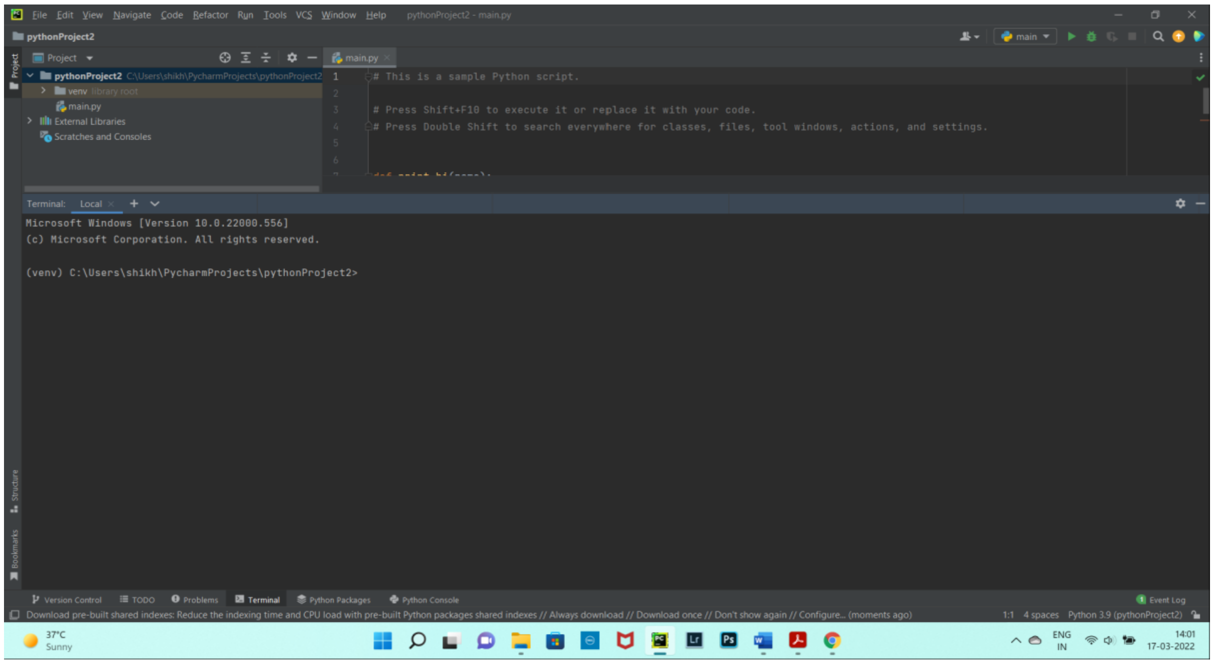
We have implemented our project in python, focusing mainly on one of its library Streamlit.

Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers. Data scientists or machine learning engineers are not web developers and they're not interested in spending weeks learning to use these frameworks tobuild web apps. Instead, they want a tool that is easier to learn and to use, as long as it can display data and collect needed parameters for modeling. Streamlit allows you to create a stunning-looking application with only a few lines of code.

5.1.1)How to use Streamlit

- **Install Streamlit**

  - Install Pycharm and create your environment
  - Open the terminal

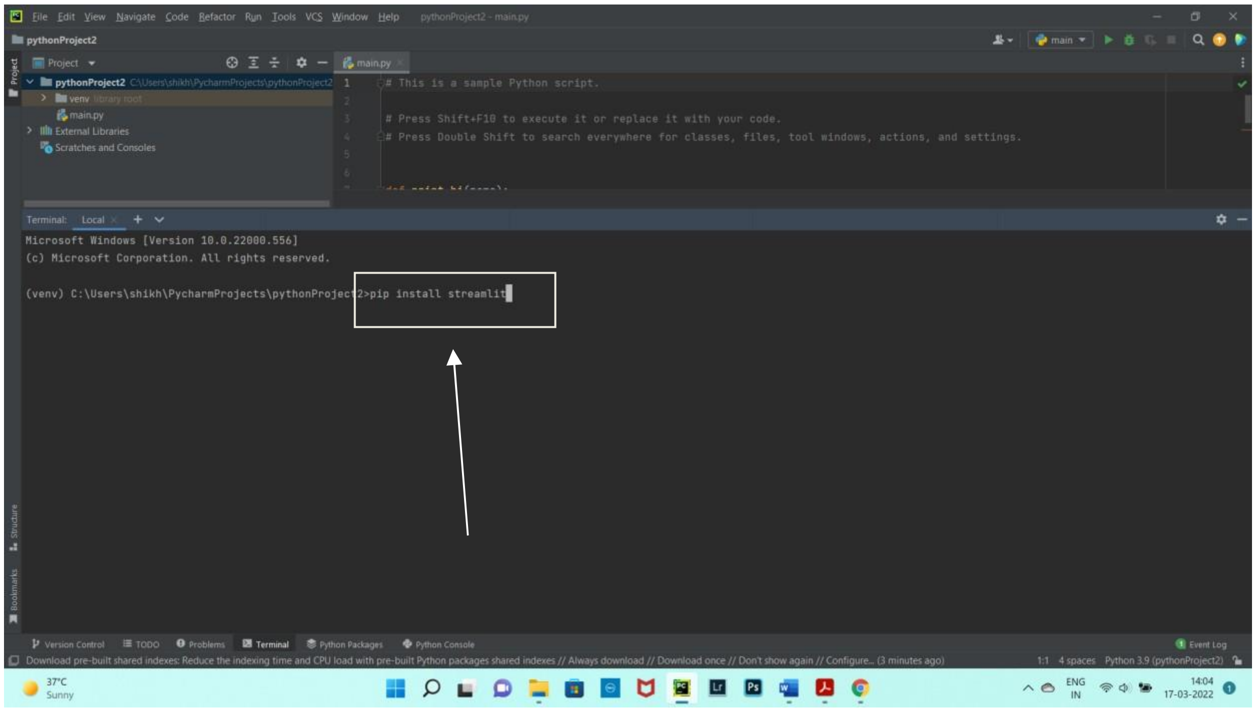


```
# This is a sample Python script.

# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
```

```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

(venv) C:\Users\shikh\PycharmProjects\pythonProject2>
```

Fig 5.1: Pycharm Terminal

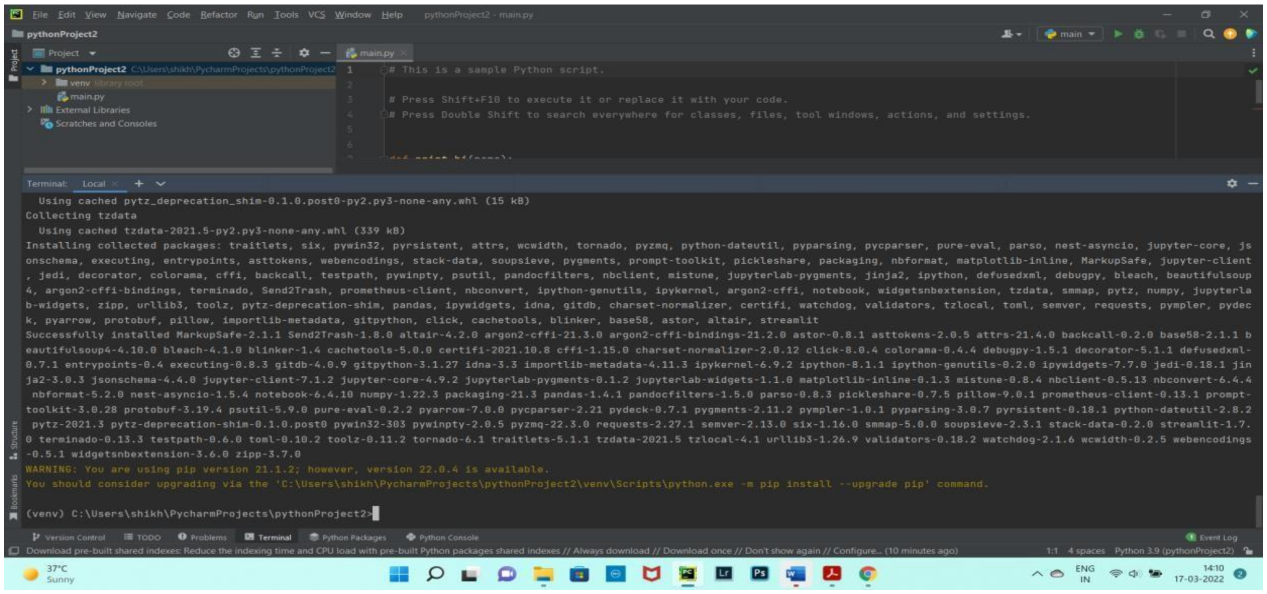
 Type this command in the terminal to install Streamlit:pip install Streamlit



```
# This is a sample Python script.

# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
```

```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

(venv) C:\Users\shikh\PycharmProjects\pythonProject2>pip install streamlit
```

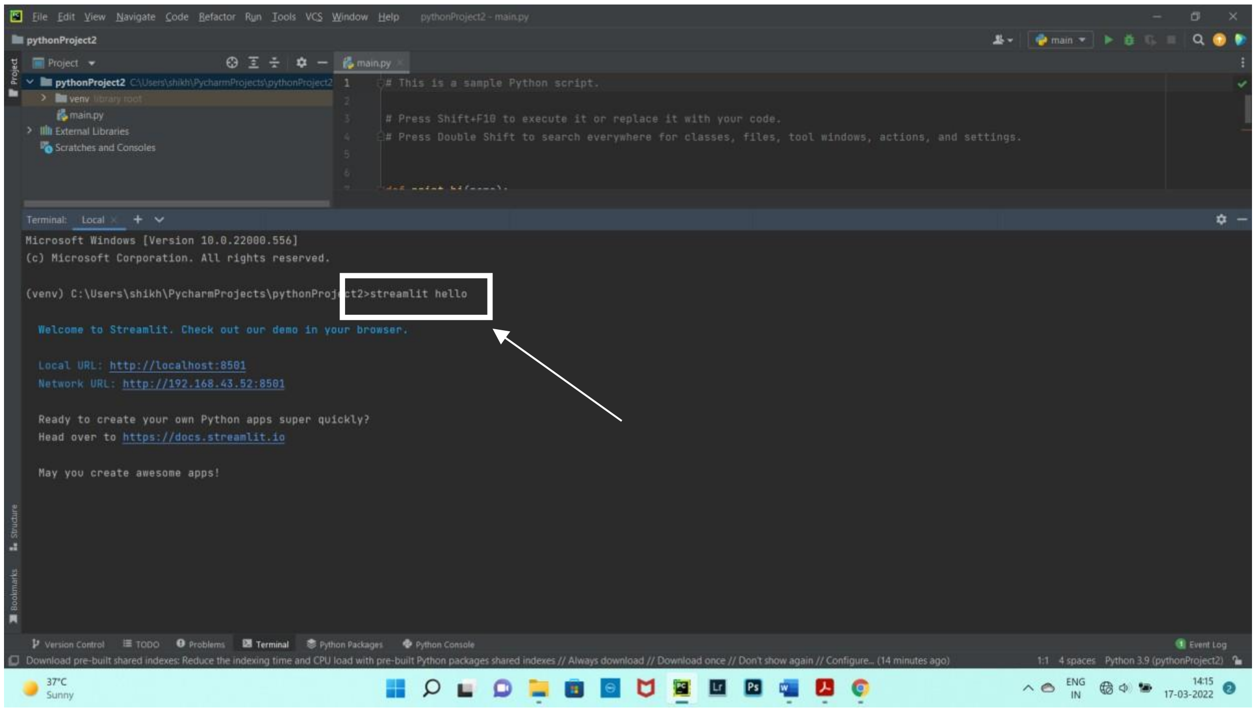
Fig 5.2: Installing Streamlit



```
  Using cached pytz_deprecation_shim-0.1.0.post0-py2.py3-none-any.whl (15 kB)
Collecting tzdata
  Using cached tzdata-2021.5-py2.py3-none-any.whl (339 kB)
Installing collected packages: traitlets, six, pywin32, pyrsistent, attrs, wcwidth, tornado, pyzmq, python-dateutil, pyparsing, pycparser, pure-eval, parso, nest-asyncio, jupyter-core, js
onschema, executing, entrypoints, asttokens, webencodings, stack-data, soupsieve, pygments, prompt-toolkit, pickleshare, packaging, nbformat, matplotlib-inline, MarkupSafe, jupyter-client
, jedi, decorator, colorama, cffi, backcall, testpath, pywinpty, psutil, pandocfilters, nbclient, mistune, jupyterlab-pygments, jinja2, ipython, defusedxml, debugpy, bleach, beautifulsoup
4, argon2-cffi-bindings, terminado, Send2Trash, prometheus-client, nbconvert, ipython-genutils, ipykernel, argon2-cffi, notebook, widgetsnbextension, tzdata, smmap, pytz, numpy, jupyterla
b-widgets, zipp, urllib3, toolz, pytz-deprecation-shim, pandas, ipywidgets, idna, gitdb, charset-normalizer, certifi, watchdog, validators, tzlocal, toml, semver, requests, pympler, pydec
k, pyarrow, protobuf, pillow, importlib-metadata, gitpython, click, cachetools, blinker, base58, astor, altair, streamlit
Successfully installed MarkupSafe-2.1.1 Send2Trash-1.8.0 altair-4.2.0 argon2-cffi-21.3.0 argon2-cffi-bindings-21.2.0 astor-0.8.1 asttokens-2.0.5 attrs-21.4.0 backcall-0.2.0 base58-2.1.1 b
eautifulsoup4-4.10.0 bleach-4.1.0 blinker-1.4 cachetools-5.0.0 certifi-2021.10.8 cffi-1.15.0 charset-normalizer-2.0.12 click-8.0.4 colorama-0.4.4 debugpy-1.5.1 decorator-5.1.1 defusedxml-
0.7.1 entrypoints-0.4 executing-0.8.3 gitdb-4.0.9 gitpython-3.1.27 idna-3.3 importlib-metadata-4.11.3 ipykernel-6.9.2 ipython-8.1.1 ipython-genutils-0.2.0 ipywidgets-7.7.0 jedi-0.18.1 jin
ja2-3.0.3 jsonschema-4.4.0 jupyter-client-7.1.2 jupyter-core-4.9.2 jupyterlab-pygments-0.1.2 jupyterlab-widgets-1.1.0 matplotlib-inline-0.1.3 mistune-0.8.4 nbclient-0.5.13 nbconvert-6.4.4
 nbformat-5.2.0 nest-asyncio-1.5.4 notebook-6.4.10 numpy-1.22.3 packaging-21.3 pandas-1.4.1 pandocfilters-1.5.0 parso-0.8.3 pickleshare-0.7.5 pillow-9.0.1 prometheus-client-0.13.1 prompt-
toolkit-3.0.28 protobuf-3.19.4 psutil-5.9.0 pure-eval-0.2.2 pyarrow-7.0.0 pycparser-2.21 pydeck-0.7.1 pygments-2.11.2 pympler-1.0.1 pyparsing-3.0.7 pyrsistent-0.18.1 python-dateutil-2.8.2
 pytz-2021.3 pytz-deprecation-shim-0.1.0.post0 pywin32-303 pywinpty-2.0.5 pyzmq-22.3.0 requests-2.27.1 semver-2.13.0 six-1.16.0 smmap-5.0.0 soupsieve-2.3.1 stack-data-0.2.0 streamlit-1.7.
0 terminado-0.13.3 testpath-0.6.0 toml-0.10.2 toolz-0.11.2 tornado-6.1 traitlets-5.1.1 tzdata-2021.5 tzlocal-4.1 urllib3-1.26.9 validators-0.18.2 watchdog-2.1.6 wcwidth-0.2.5 webencodings
-0.5.1 widgetsnbextension-3.6.0 zipp-3.7.0
WARNING: You are using pip version 21.1.2; however, version 22.0.4 is available.
You should consider upgrading via the 'C:\Users\shikh\PycharmProjects\pythonProject2\venv\Scripts\python.exe -m pip install --upgrade pip' command.

(venv) C:\Users\shikh\PycharmProjects\pythonProject2>
```

Fig 5.3: Streamlit installed

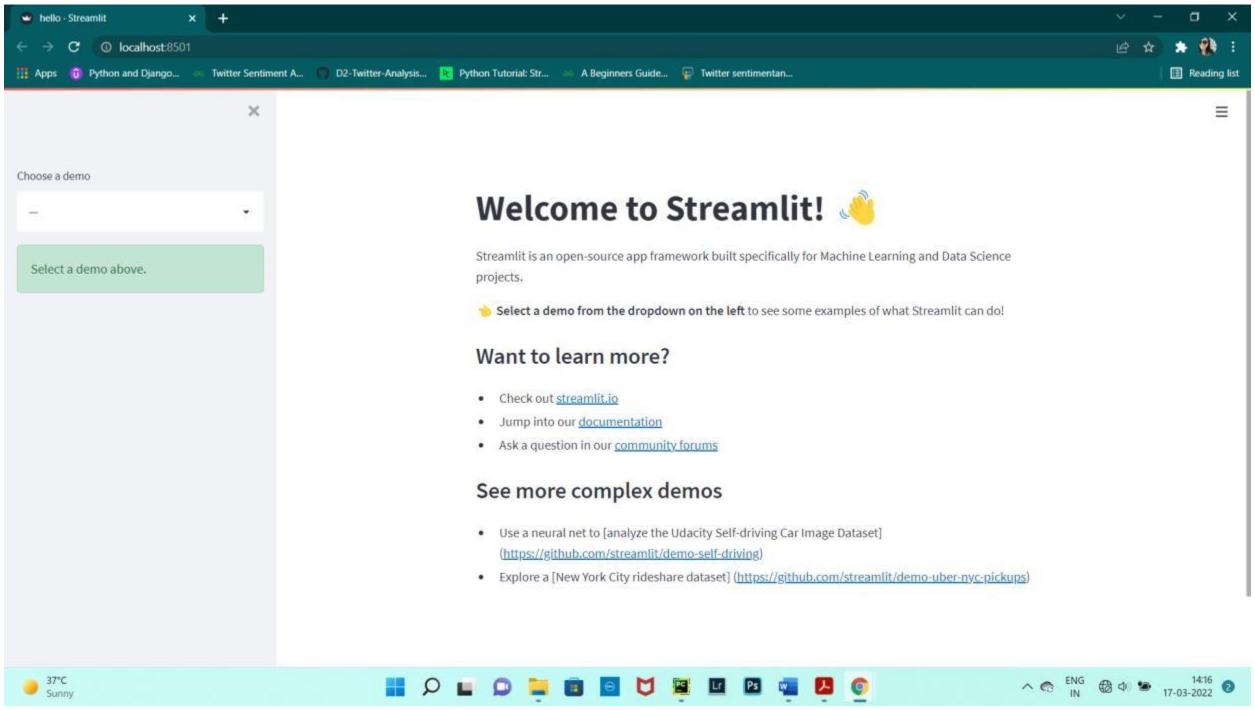
□ Test if the installation worked:



```
# This is a sample Python script.

# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
```

```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

(venv) C:\Users\shikh\PycharmProjects\pythonProject2>streamlit hello

  Welcome to Streamlit. Check out our demo in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.43.52:8501

  Ready to create your own Python apps super quickly?
  Head over to https://docs.streamlit.io

  May you create awesome apps!
```

Fig 5.4: Installation check



Choose a demo

—

Select a demo above.

Welcome to Streamlit! 👋

Streamlit is an open-source app framework built specifically for Machine Learning and Data Science projects.

👈 **Select a demo from the dropdown on the left** to see some examples of what Streamlit can do!

Want to learn more?

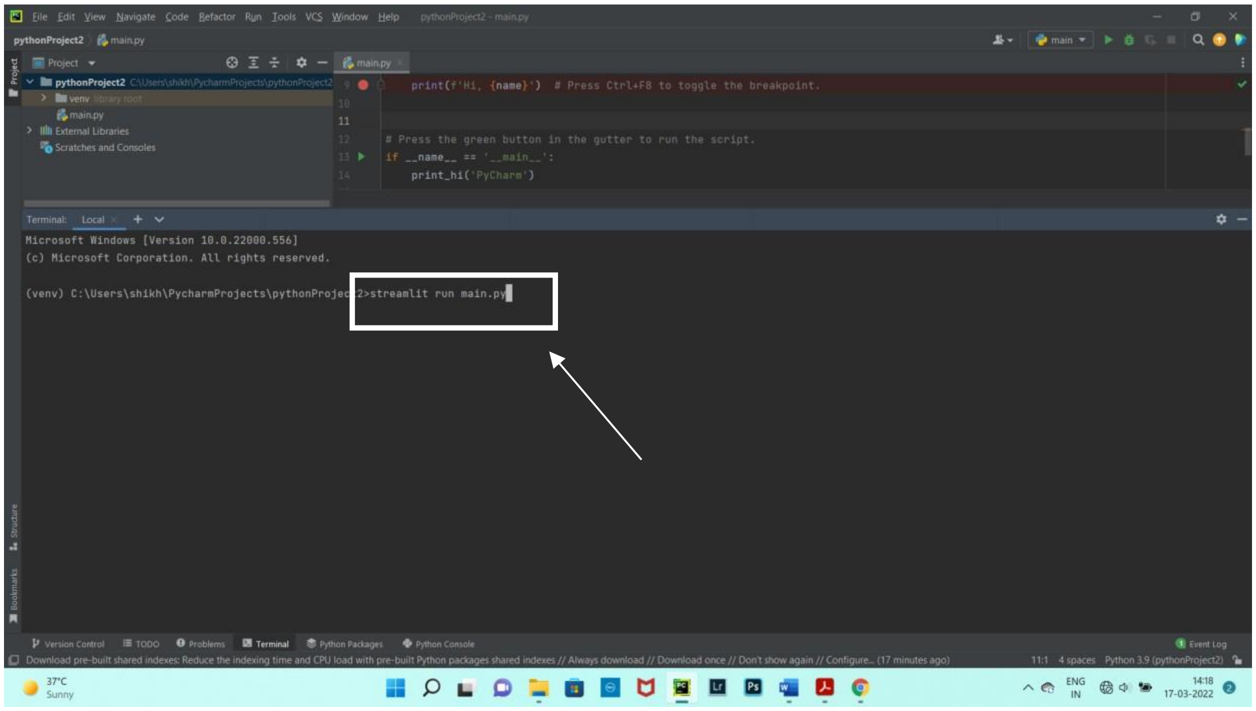
- Check out streamlit.io
- Jump into our documentation
- Ask a question in our community forums

See more complex demos

- Use a neural net to [analyze the Udacity Self-driving Car Image Dataset] (https://github.com/streamlit/demo-self-driving)
- Explore a [New York City rideshare dataset] (https://github.com/streamlit/demo-uber-nyc-pickups)

Fig 5.5: User Interface of Webapp

How to run your Streamlit code:



```
    print(f'Hi, {name}')  # Press Ctrl+F8 to toggle the breakpoint.


# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')
```

```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

(venv) C:\Users\shikh\PycharmProjects\pythonProject2>streamlit run main.py
```

Fig 5.6: Executing main file

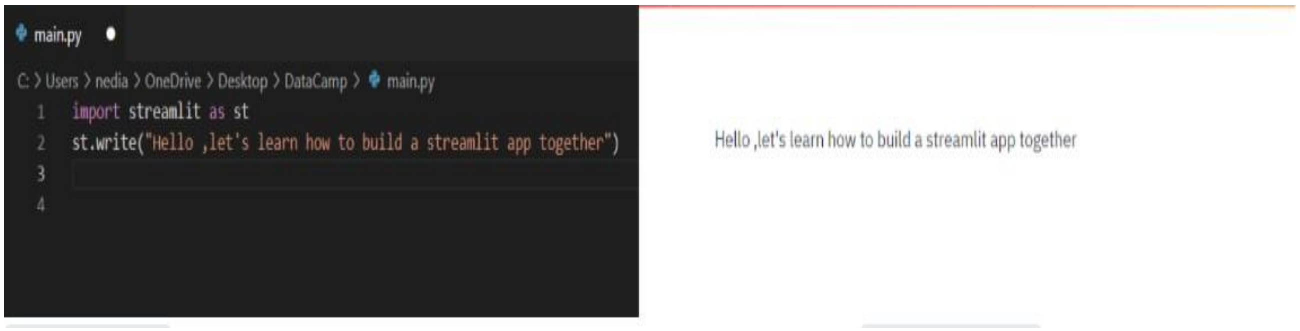
Streamlit commands are easy to write and understand. With just a simple command, you are ableto display texts, media, widgets, graphs, etc.

Display texts with Streamlit

In the beginning, we will see how to add text to your Streamlit app, and what the differentcommands are to add texts.

st.write(): This function is used to add anything to a web app, from formatted string to charts inmatplotlib figure, Altair charts, plotly figure, data frame, Keras model, and others.

```
import streamlit as st
st.write("Hello ,let's learn how to build a streamlit app together")
```



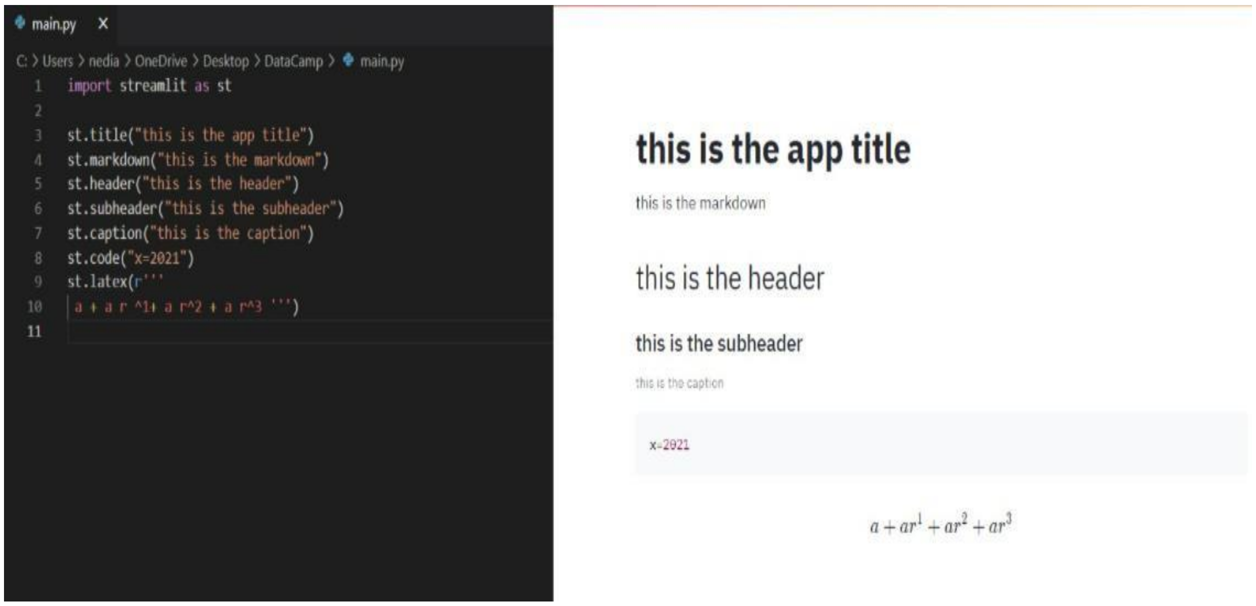
```
main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py
import streamlit as st
st.write("Hello ,let's learn how to build a streamlit app together")
```

```
Hello ,let's learn how to build a streamlit app together
```

Fig 5.7: main file

st.title(): This function allows you to add the title of the app. st.header(): This function is used toset header of a section. st.markdown(): This function is used to set a markdown of a section. st.subheader(): This function is used to set sub-header of a section. st.caption(): Thisfunction is used to write caption. st.code(): This function is used to set a code. st.latex(): This function is used to display mathematical expressions formatted as LaTeX.

```
st.title ("this is the app title")
st.header("this is the markdown")
st.markdown("this is the header")
st.subheader("this is the subheader")
st.caption("this is the caption")
st.code("x=2021")
st.latex(r''' a+a r^1+a r^2+a r^3 ''')
```



main.py

C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py

```python
import streamlit as st

st.title("this is the app title")
st.markdown("this is the markdown")
st.header("this is the header")
st.subheader("this is the subheader")
st.caption("this is the caption")
st.code("x=2021")
st.latex(r'''
 a + a r ^1+ a r^2 + a r^3 ''')
```

this is the app title

this is the markdown

this is the header

this is the subheader

this is the caption

```
x=2021
```

$$a + ar^1 + ar^2 + ar^3$$

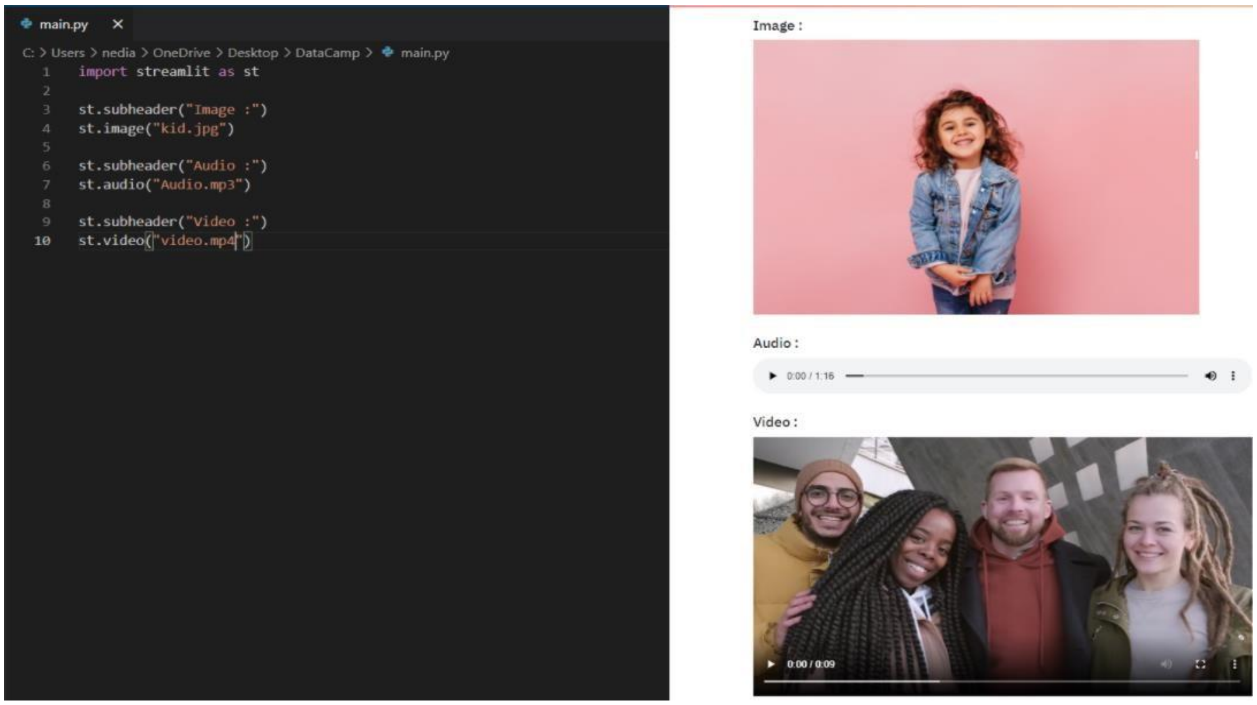
Fig 5.8: Title

Display an image, video or audio file with Streamlit

You can't find functions as easy as Streamlit functions to display images, videos, and audio files.Let's take a look at how to display media with Streamlit !

st.image(): This function is used to display an image. st.audio(): This function is used to displayan audio. st.video(): This function is used to display a video.

```python
st.image("kid.jpg")
st.audio("Audio.mp3")
st.video("video.mp4")
```


main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py

```python
import streamlit as st

st.subheader("Image :")
st.image("kid.jpg")

st.subheader("Audio :")
st.audio("Audio.mp3")

st.subheader("Video :")
st.video("video.mp4")
```

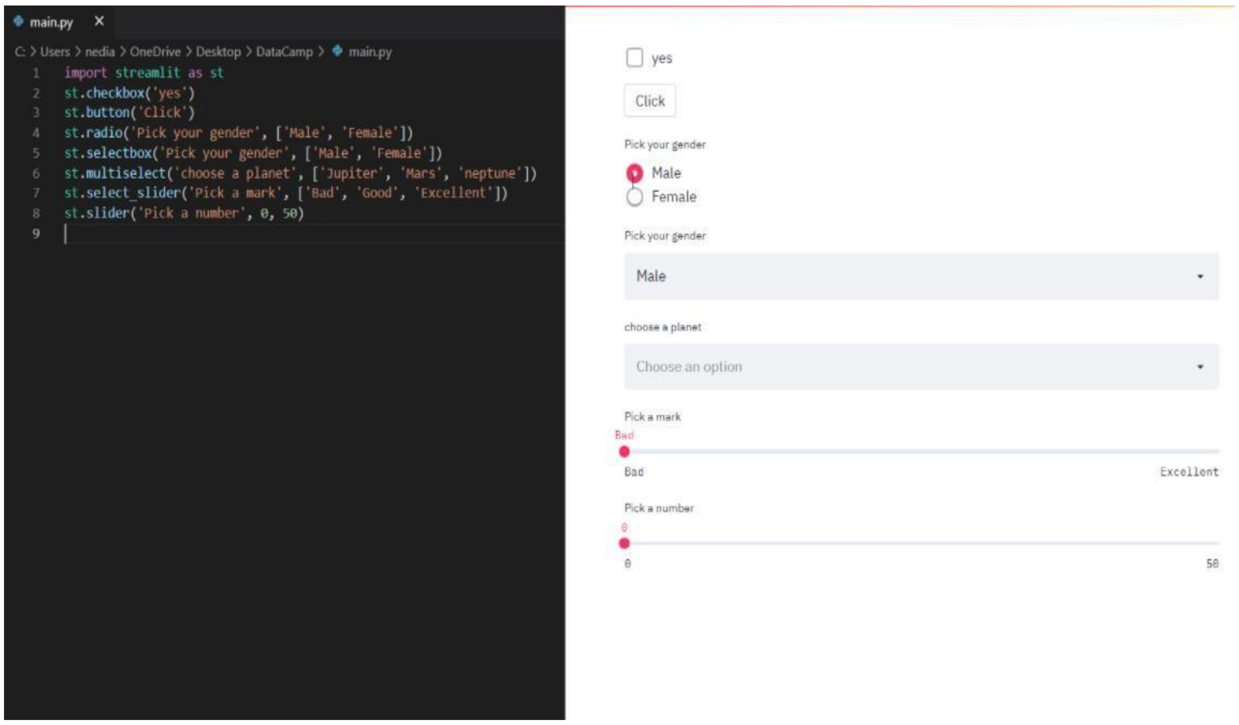
Fig 5.9: Displaying Multimedia

Input widgets

Widgets are the most important user interface components. Streamlit has various widgets thatallow you to bake interactivity directly into your apps with buttons, sliders, text inputs, and more.

st.checkbox(): This function returns a Boolean value. When the box is checked, it returns a True value, otherwise a False value. st.button(): This function is used to display a button widget. st.radio(): This function is used to display a radio button widget. st.selectbox(): Thisfunction is used to display a select widget. st.multiselect(): This function is used to display a multiselect widget. st.select_slider(): This function is used to display a select slider widget. st.slider(): This function is used to display a slider widget.

```
st.checkbox('yes')

st.button('Click')
st.radio('Pick your gender',['Male','Female'])

st.selectbox('Pick your gender',['Male','Female'])

st.multiselect('choose a planet',['Jupiter', 'Mars', 'neptune'])

st.select_slider('Pick a mark', ['Bad', 'Good', 'Excellent'])

st.slider('Pick a number', 0,50)
```

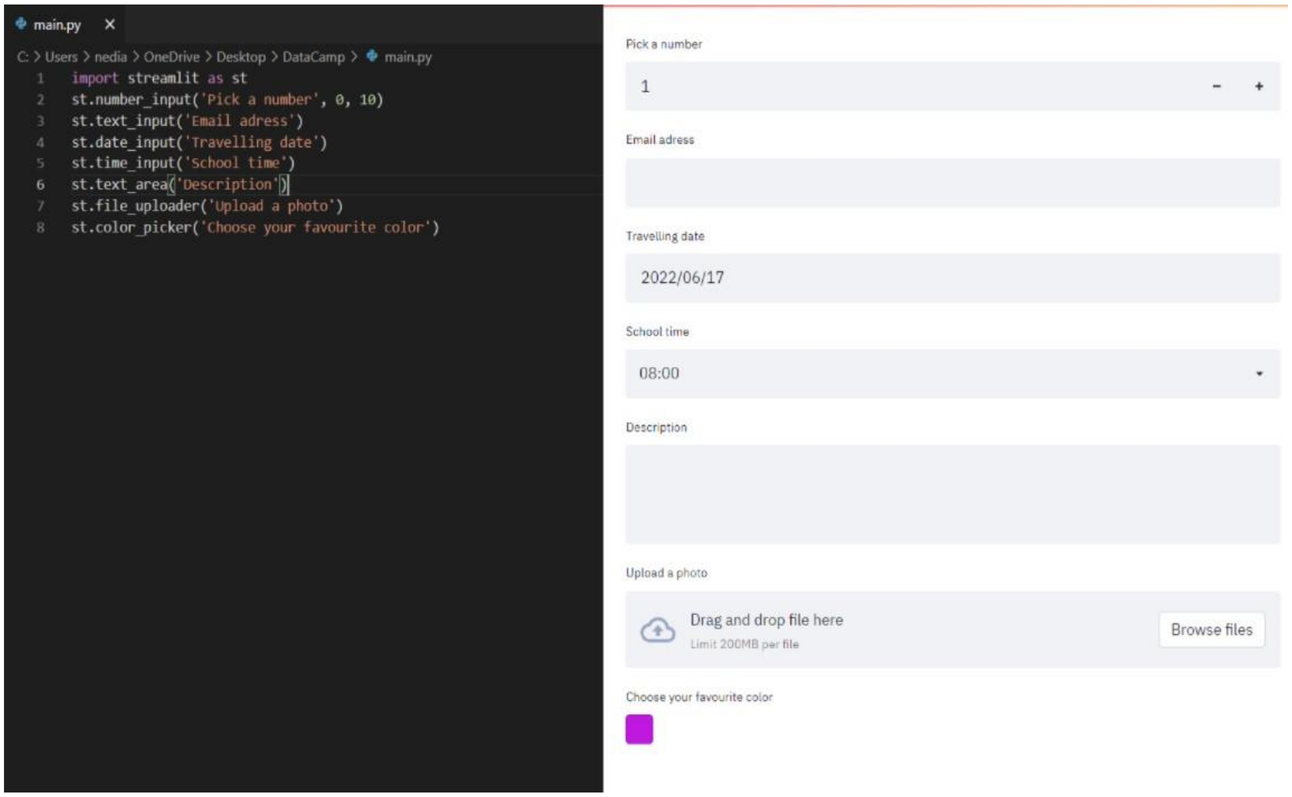


```
import streamlit as st
st.checkbox('yes')
st.button('Click')
st.radio('Pick your gender', ['Male', 'Female'])
st.selectbox('Pick your gender', ['Male', 'Female'])
st.multiselect('choose a planet', ['Jupiter', 'Mars', 'neptune'])
st.select_slider('Pick a mark', ['Bad', 'Good', 'Excellent'])
st.slider('Pick a number', 0, 50)
```

Fig 5.10 Input Widgets

st.number_input(): This function is used to display a numeric input widget. st.text_input(): This function is used to display a text input widget. st.date_input(): This function is used to display a date input widget to choose a date. st.time_input(): This function is used to display a time input widget to choose a time. st.text_area(): This function is used to display a text input widget with more than a line text. st.file_uploader(): This function is used to display a file uploader widget. st.color_picker(): This function is used to display color picker widget to choose a color.

```
st.number_input('Pick a number', 0,10)
st.text_input('Email address')
st.date_input('Travelling date')
st.time_input('School time')
st.text_area('Description')
st.file_uploader('Upload a photo')
st.color_picker('Choose your favorite color')
```



```
import streamlit as st
st.number_input('Pick a number', 0, 10)
st.text_input('Email adress')
st.date_input('Travelling date')
st.time_input('School time')
st.text_area('Description')
st.file_uploader('Upload a photo')
st.color_picker('Choose your favourite color')
```

Pick a number: 1

Email adress

Travelling date: 2022/06/17

School time: 08:00

Description

Upload a photo

main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py
Drag and drop file here
Limit 200MB per file
Browse files
Choose your favourite color

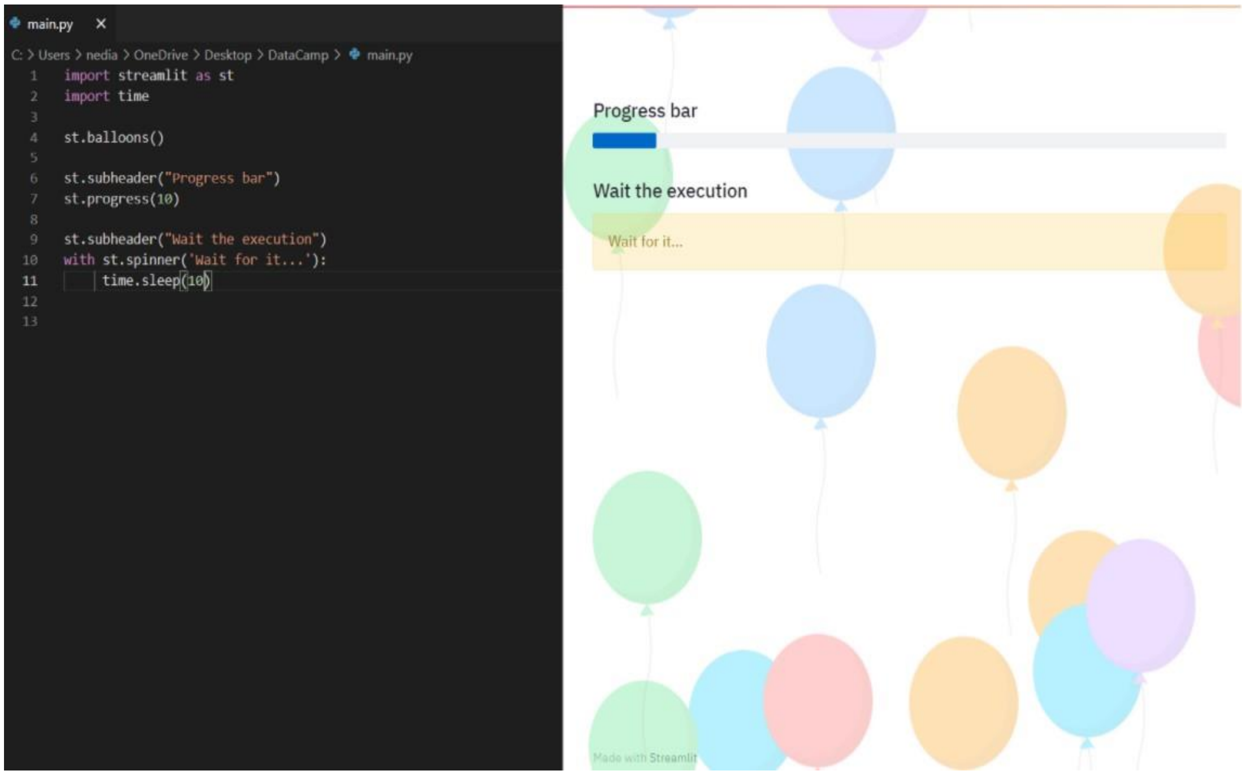
Fig 5.11 Displaying number input

Display progress and status with Streamlit

Now we will see how we can add a progress bar and status messages such as error and success toour app.

st.balloons(): This function is used to display balloons for celebration. st.progress(): Thisfunction is used to display a progress bar. st.spinner(): This function is used to display a temporary waiting message during execution.

```
st.balloons()
st.progress(10)
with st.spinner('Wait for it...'):
    time.sleep(10)
```



```
import streamlit as st
import time

st.balloons()

st.subheader("Progress bar")
st.progress(10)

st.subheader("Wait the execution")
with st.spinner('Wait for it...'):
    time.sleep(10)
```

Progress bar

Wait the execution

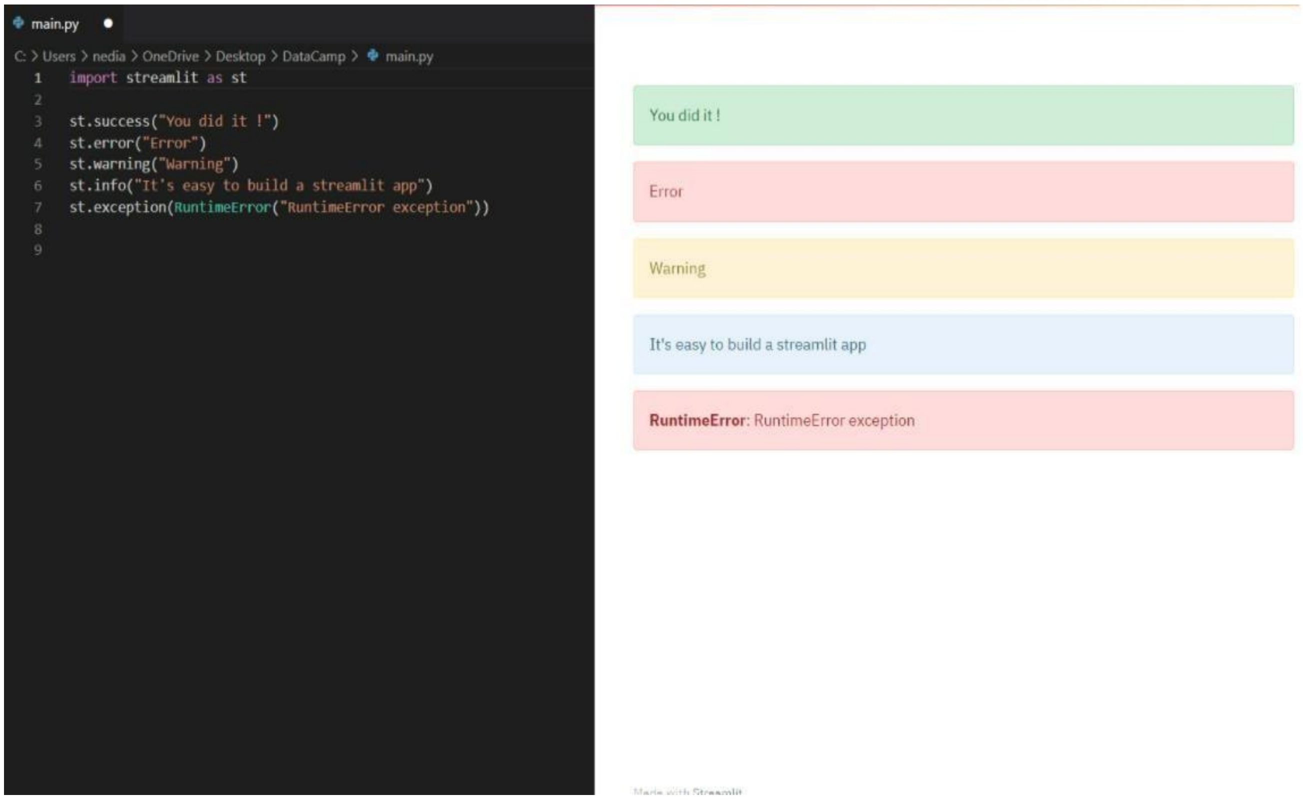
Wait for it...

main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py
Made with Streamlit

Fig 5.12: Displaying Progress

st.success(): This function is used to display a success message. st.error(): This function is usedto display an error message. st.warning(): This function is used to display a warning message. st.info(): This function is used to display an informational message. st.exception(): Thisfunction is used to display an exception message.

```
st.success("You did it !")

st.error("Error")

st.warnig("Warning")

st.info("It's easy to build a streamlit app")

st.exception(RuntimeError("RuntimeError exception"))
```



```
import streamlit as st

st.success("You did it !")
st.error("Error")
st.warning("Warning")
st.info("It's easy to build a streamlit app")
st.exception(RuntimeError("RuntimeError exception"))
```

You did it !

Error

Warning

It's easy to build a streamlit app

RuntimeError: RuntimeError exception

main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py
Made with Streamlit

Fig 5.13: Displaying Status

Sidebar and container

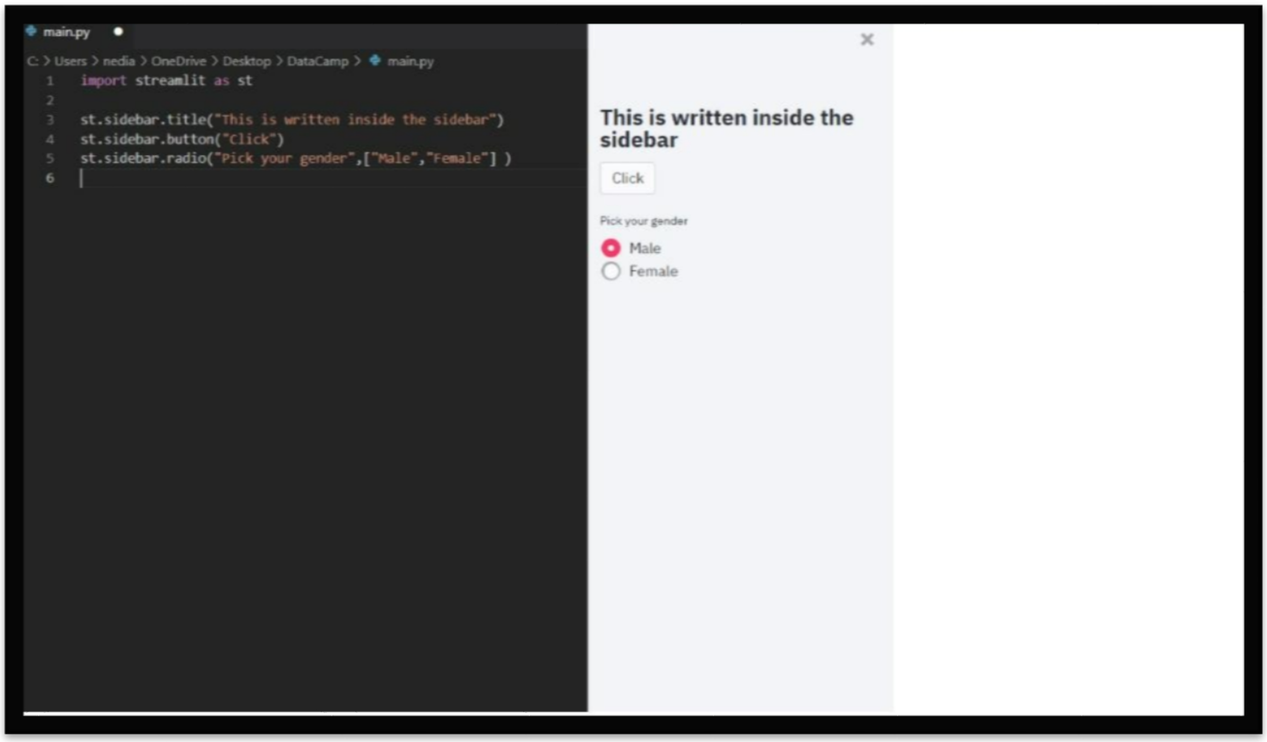
You can also create a sidebar or a container on your page to organize your app. The hierarchyand arrangement of pages on your app can have a large impact on your user experience. By organizing your content, you allow visitors to understand and navigate your site, which helps them find what they're looking for and increases the likelihood that they'll return in the future.

- **Sidebar**

Passing an element to st.sidebar() will make this element pinned to the left, allowing users tofocus on the content in your app.

But st.spinner() and st.echo() are not supported with st.sidebar.

As you see, you can create a sidebar in your app interface and put elements inside it that willmake your app more organized and easier to understand.



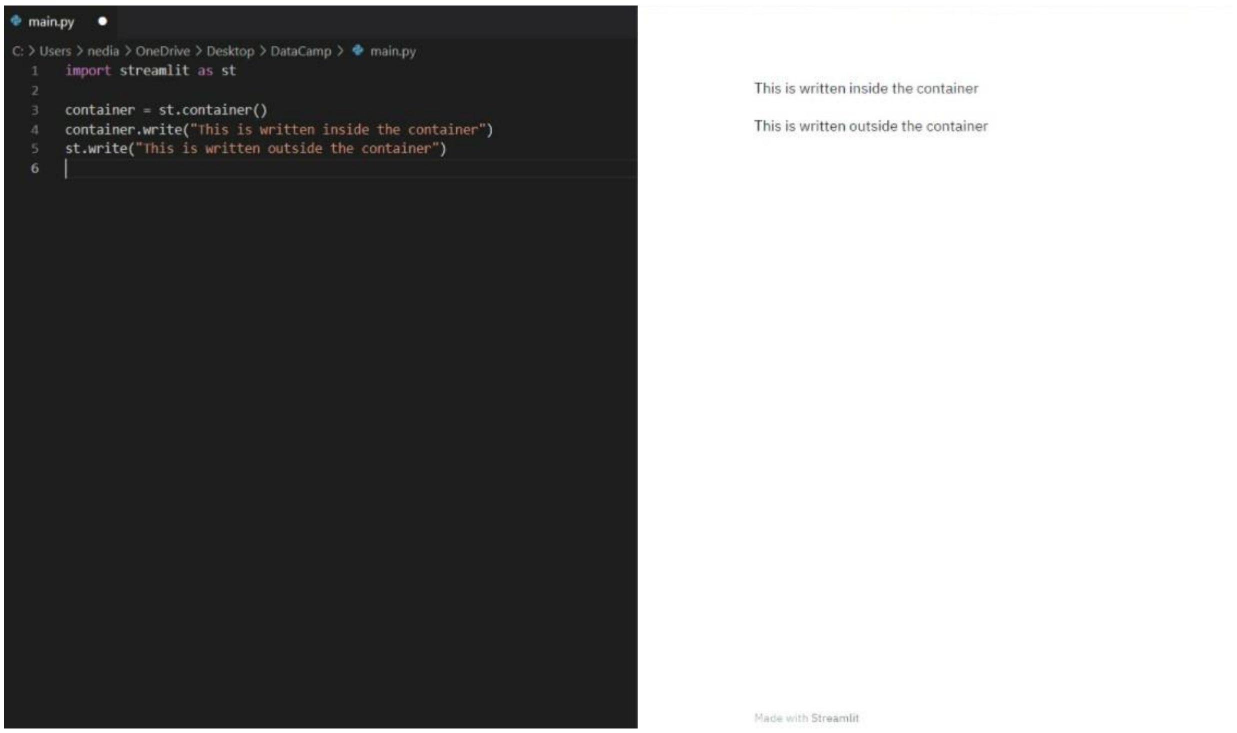
```
import streamlit as st

st.sidebar.title("This is written inside the sidebar")
st.sidebar.button("Click")
st.sidebar.radio("Pick your gender",["Male","Female"] )
```

main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py
This is written inside the sidebar
Click
Pick your gender
Male
Female

Fig 5.14: Sidebar

- **Container**

st.container() is used to create an invisible container where you can put elements in order tocreate a useful arrangement and hierarchy.



```python
import streamlit as st

container = st.container()
container.write("This is written inside the container")
st.write("This is written outside the container")
```

```
This is written inside the container

This is written outside the container
```

main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py
Made with Streamlit

Fig 5.15 Container

Display graphs with Streamlit

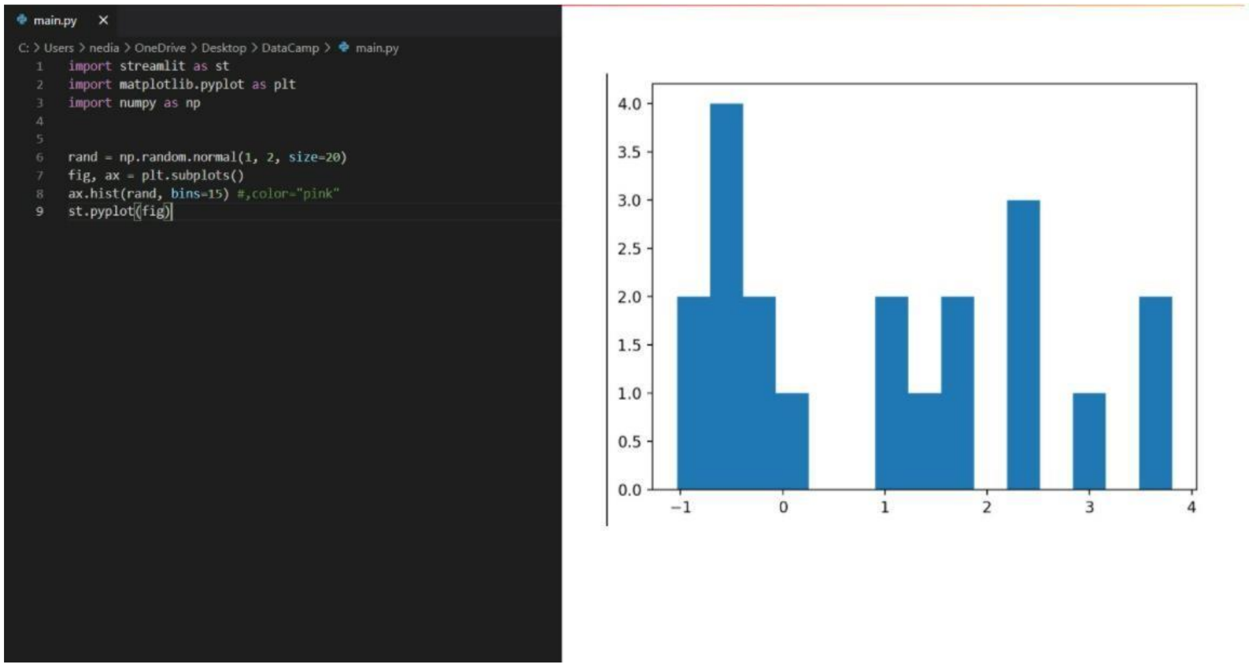
- **Why do we need visualization?**

Data visualization helps to tell stories by curating data into a format that's easier to understand, highlighting the trends and outliers. A good visualization tells a story, removing the noise from data and highlighting the useful information. However, it's not simply as easy as dressing up a graph to make it look better or slapping on the "info" part of an infographic. Effective data visualization is a delicate balancing act between form and function. The plainest graph could be too boring to draw attention or convey a powerful message, and the most stunning visualization could utterly fail at conveying the right message. The data and the visuals need to work together,and there's an art to combining great analysis with great storytelling.

Do you think giving you the data of one million points in a table/database file and asking you to provide your inferences by just seeing the data on that table is feasible? Unless you're a super human, it's not possible. This is when we make use of data visualization—it gives us a clear ideaof what the information means by giving it visual context through maps or graphs. That's the power of Streamlit visualization.

st.pyplot(): This function is used to display a matplotlib.pyplot figure.

```
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np

rand=np.random.normal(1, 2, size=20)
fig, ax = plt.subplots()
ax.hist(rand, bins=15)
st.pyplot(fig)
```

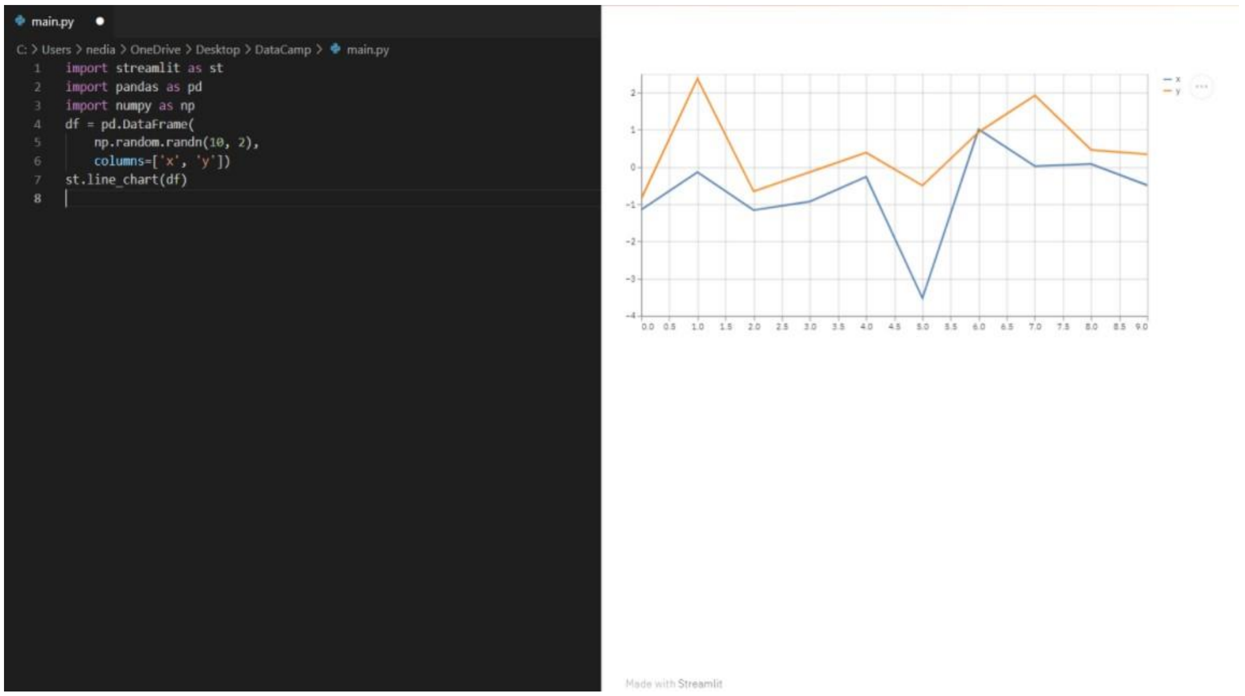

main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−1
0
1
2
3
4

```
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np


rand = np.random.normal(1, 2, size=20)
fig, ax = plt.subplots()
ax.hist(rand, bins=15) #,color="pink"
st.pyplot(fig)
```

Fig 5.16 :Displaying Graph

st.line_chart(): This function is used to display a line chart.

```
import streamlit as st
import pandas as pd
import numpy as np
df= pd.DataFrame(
    np.random.randn(10, 2),
    columns=['x', 'y'])
st.line_chart(df)
```

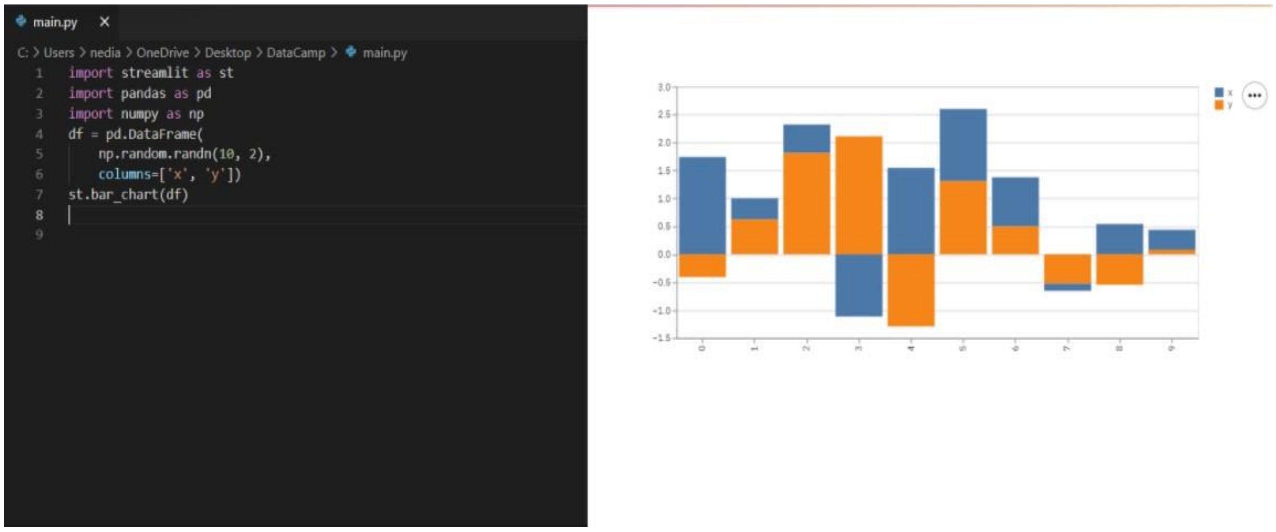


```
import streamlit as st
import pandas as pd
import numpy as np
df = pd.DataFrame(
    np.random.randn(10, 2),
    columns=['x', 'y'])
st.line_chart(df)
```

Fig 5.17: Line chart

st.bar_chart(): This function is used to display a bar chart.

```
import streamlit as st
import pandas as pd
import numpy as np
df= pd.DataFrame(
    np.random.randn(10, 2),
    columns=['x', 'y'])
st.bar_chart(df)
```


main.py

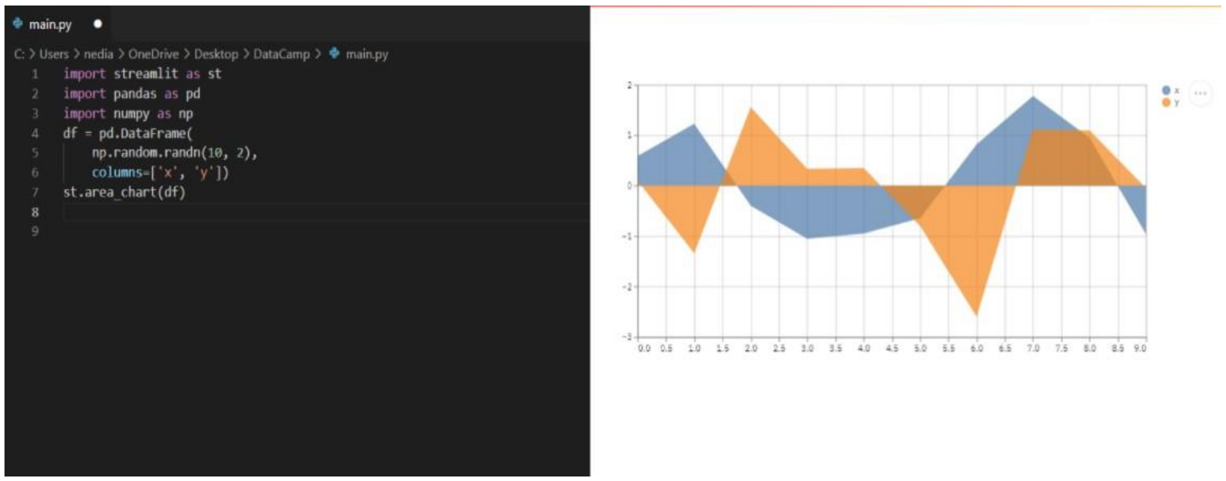
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py

```python
import streamlit as st
import pandas as pd
import numpy as np
df = pd.DataFrame(
    np.random.randn(10, 2),
    columns=['x', 'y'])
st.bar_chart(df)
```

Fig 5.18: Bar Chart

st.area_chart(): This function is used to display an area chart.

```python
import streamlit as st
import pandas as pd
import numpy as np
df= pd.DataFrame(
    np.random.randn(10, 2),
    columns=['x', 'y'])
```


main.py

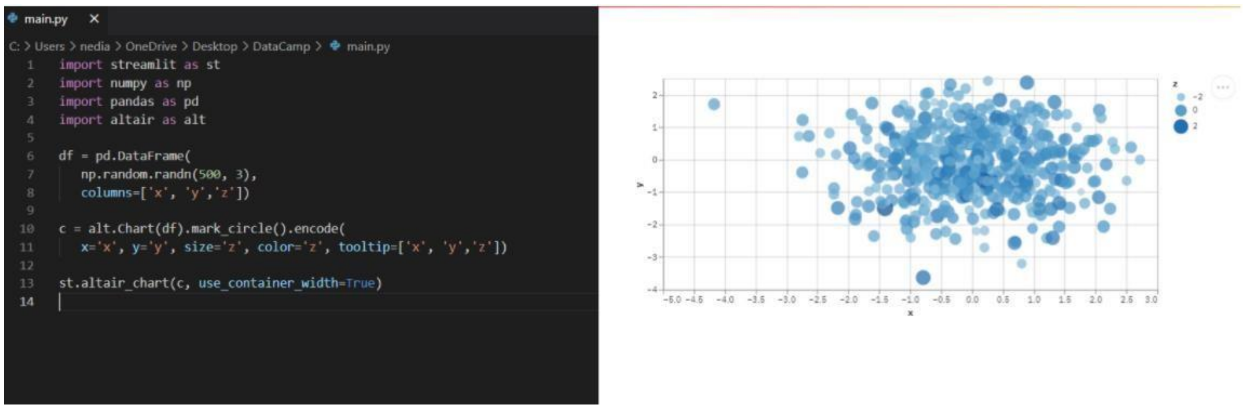
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py

```python
import streamlit as st
import pandas as pd
import numpy as np
df = pd.DataFrame(
    np.random.randn(10, 2),
    columns=['x', 'y'])
st.area_chart(df)
```

Fig 5.19: Area Chart

st.altair_chart(): This function is used to display an altair chart.

```
import streamlit as st
import numpy as np
import pandas as pd
import altair as alt

df = pd.DataFrame(
  np.random.randn(500, 3),
  columns=['x','y','z'])

c = alt.Chart(df).mark_circle().encode(
  x='x' , 'y'=y , size='z', color='z', tooltip=['x', 'y', 'z'])
st.altair_chart(c, use_container_width=True)
```



main.py

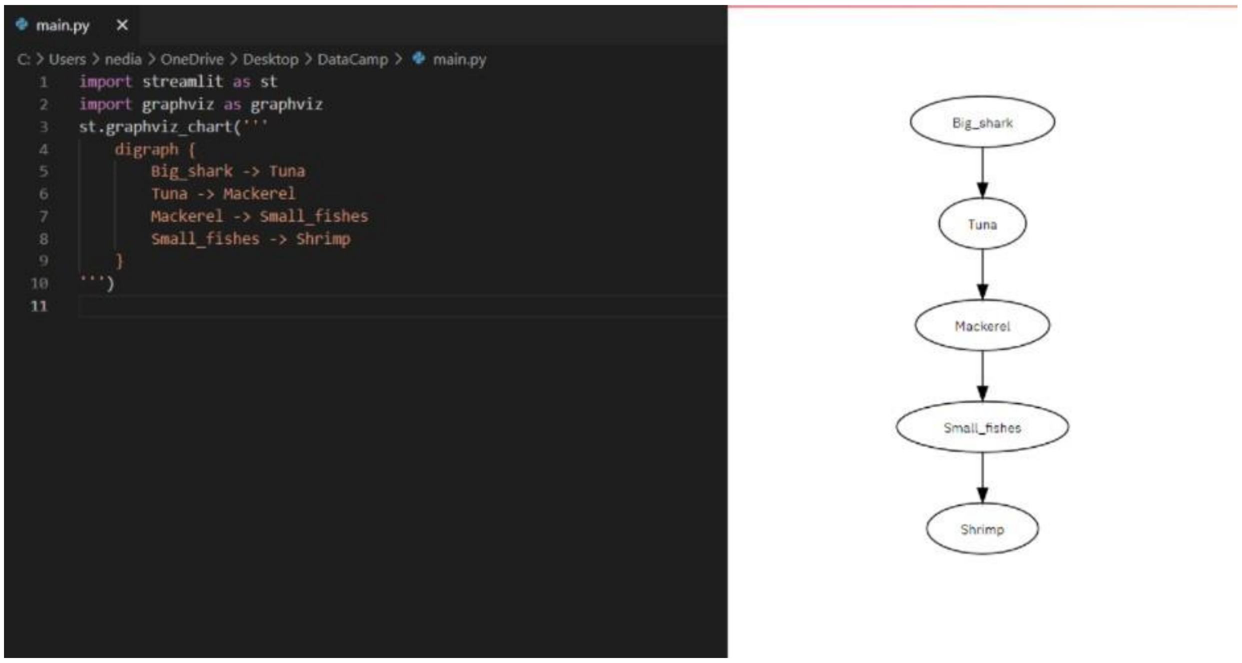
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py

```
import streamlit as st
import numpy as np
import pandas as pd
import altair as alt

df = pd.DataFrame(
    np.random.randn(500, 3),
    columns=['x', 'y','z'])

c = alt.Chart(df).mark_circle().encode(
    x='x', y='y', size='z', color='z', tooltip=['x', 'y','z'])

st.altair_chart(c, use_container_width=True)
```

Fig 5.20 Altair chart

st.graphviz_chart(): This function is used to display graph objects, which can be completed usingdifferent nodes and edges.

```
import streamlit as st
import graphviz as graphviz
st.graphviz_chart('''
    digraph {
        Big_shark -> Tuna
        Tuna -> Mackerel
        Mackerel -> Small_fishes
        Small_fishes -> Shrimp
    }
''')
```


main.py
C: > Users > nedia > OneDrive > Desktop > DataCamp > main.py

```
import streamlit as st
import graphviz as graphviz
st.graphviz_chart('''
    digraph {
        Big_shark -> Tuna
        Tuna -> Mackerel
        Mackerel -> Small_fishes
        Small_fishes -> Shrimp
    }
''')
```

Fig 5.21: Graphviz chart

5.2) SOFTWARE TESTING:

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –
undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last
chance to detect and correct errors before the system is installed for user acceptance testing.

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –
undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the lastchance to detect and correct errors before the system is installed for user acceptance testing
tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business processperforms accurately to the documented specifications and contains clearly definedinputs and expected results.

5.2.1) INTEGRATION TESTING:

Integration tests are designed to test integrated software components to
determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shownby successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.2.2) FUNCTIONAL TEST:

Functional tests provide systematic demonstrations that functions tested areavailable as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised

Systems/Procedures: interfacing systems or procedures must be invoked.

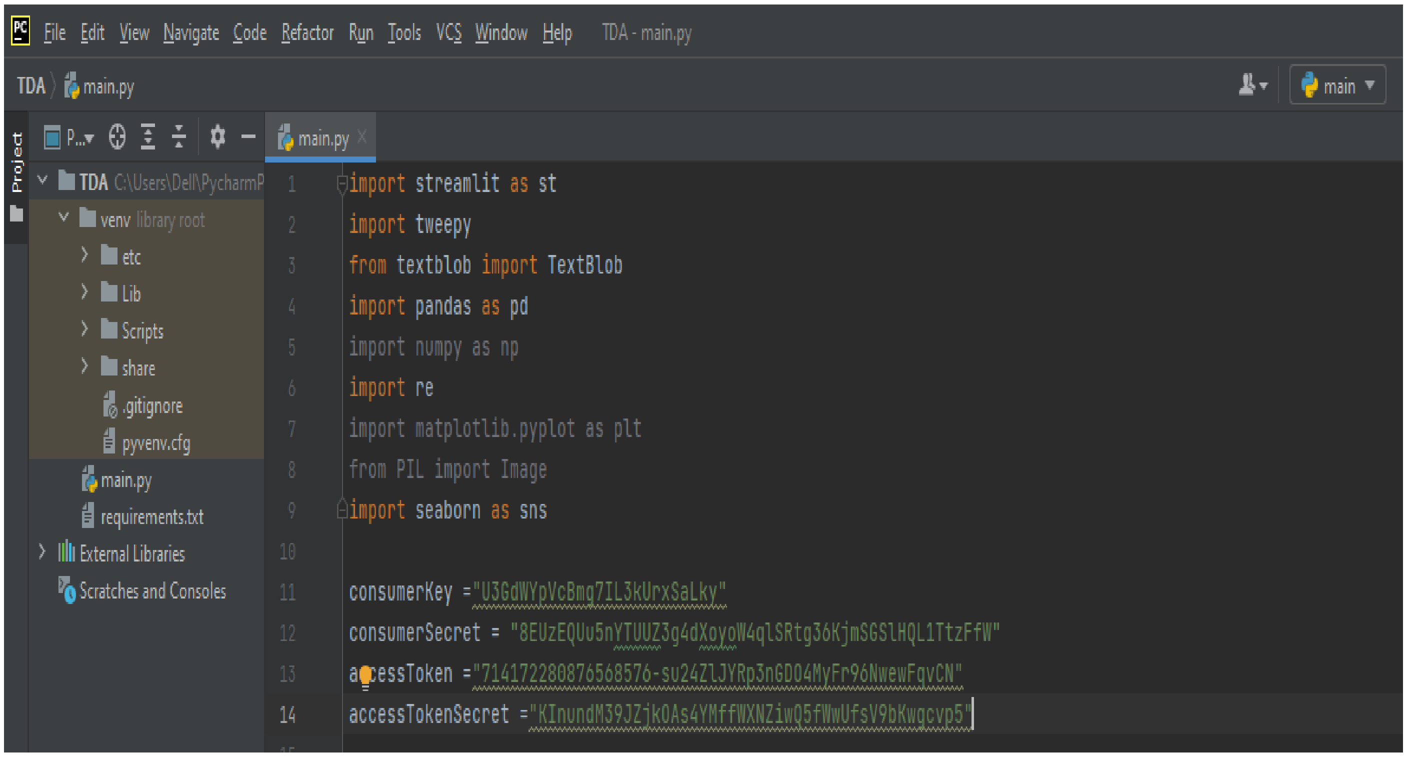
Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.2.3) SYSTEM TESTING:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. Anexample of system testing is the configuration oriented system integration test.
System testing is based on process descriptions and flows, emphasizing pre-drivenprocess links and integration points

5.3) CODE:

5.3.1) Installing and importing all necessary libraries by creating a .py file.



```python
import streamlit as st
import tweepy
from textblob import TextBlob
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
from PIL import Image
import seaborn as sns

consumerKey ="U3GdWYpVcBmg7IL3kUrxSaLky"
consumerSecret = "8EUzEQUu5nYTUUZ3g4dXoyoW4qlSRtg36KjmSGSlHQL1TtzFfW"
accessToken ="714172280876568576-su24ZlJYRp3nGD04MyFr96NwewFqvCN"
accessTokenSecret ="KInundM39JZjk0As4YMffWXNZiwQ5fWwUfsV9bKwgcvp5"
```

Fig 5.22: Importing libraries

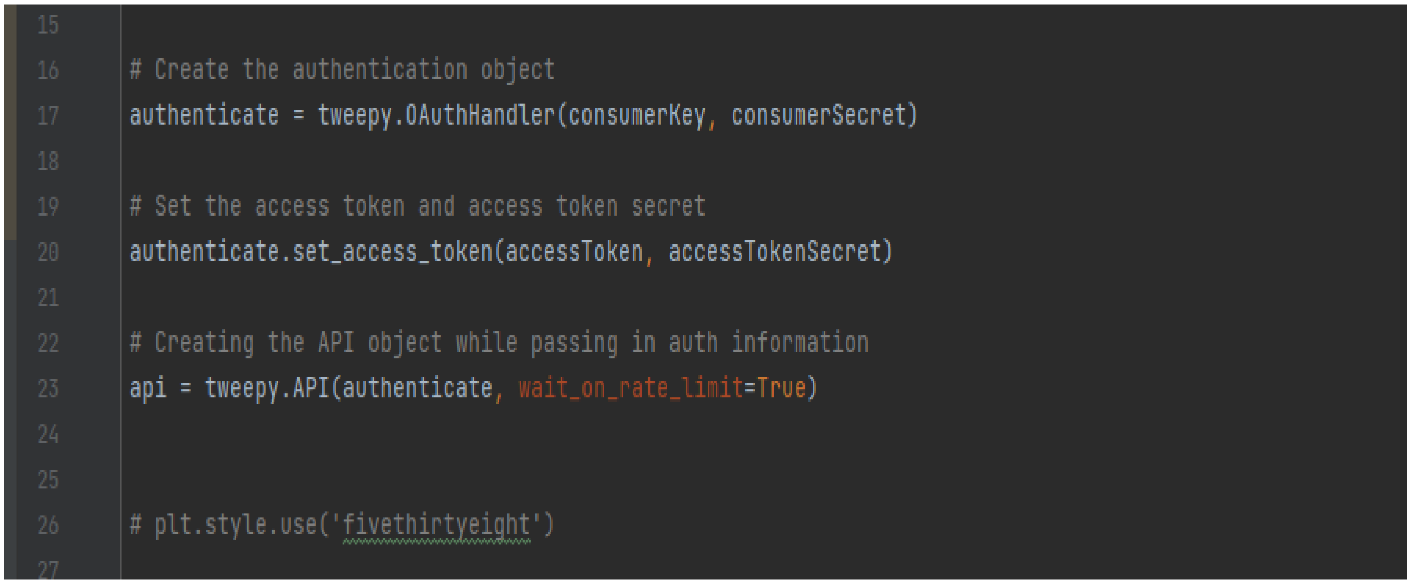
1. Streamlit: Streamlit is **an open source app framework in Python language**. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.
2. Tweepy: - Tweepy is an open source Python package that **gives you a very convenient way to access the Twitter API with Python**.
3. Textblob: - As TextBlob is a Lexicon-based sentiment analyser. **It has some predefined rules or we can say word and weight dictionary, where it has some scores that help to calculate a sentence's polarity**. That's why the Lexicon-based sentiment analyzers are also called “Rule-based sentiment analyzers”.
4. Pandas: - Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word **Panel Data**, which means **an Econometrics from Multidimensional data**. It is used for data analysis in Python and developed by **Wes McKinney** in **2008**.

5. Numpy: - NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.
6. Re : - A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).
7. Matplotlib: - It is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is **a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource**. Matplotlib uses an object oriented API to embed plots in Python applications.
8. PIL: - Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images.
9. Seaborn: - Seaborn is **a Python data visualization library based on matplotlib**. It provides a high-level interface for drawing attractive and informative statistical graphics.

- Now the first step we see in our application setup above is getting tweets from Twitter. To do that we have to access Twitters API and use it to fetch the tweets. To access Twitter API, we need API keys and API key secrets. To get them do follow the below steps

1. Create a Twitter account if you don’t have one
2. Apply for a developer account on Twitter
3. Once you get access to your developer account, start a project and get create an app instance
4. This will generate your keys. Get ‘consumer key’, ‘consumer secret’, ‘access token’, ‘access token secret’,‘consumer key’, ‘consumer secret’ are nothing but ‘API key’ and ‘API key secret’ respectively.

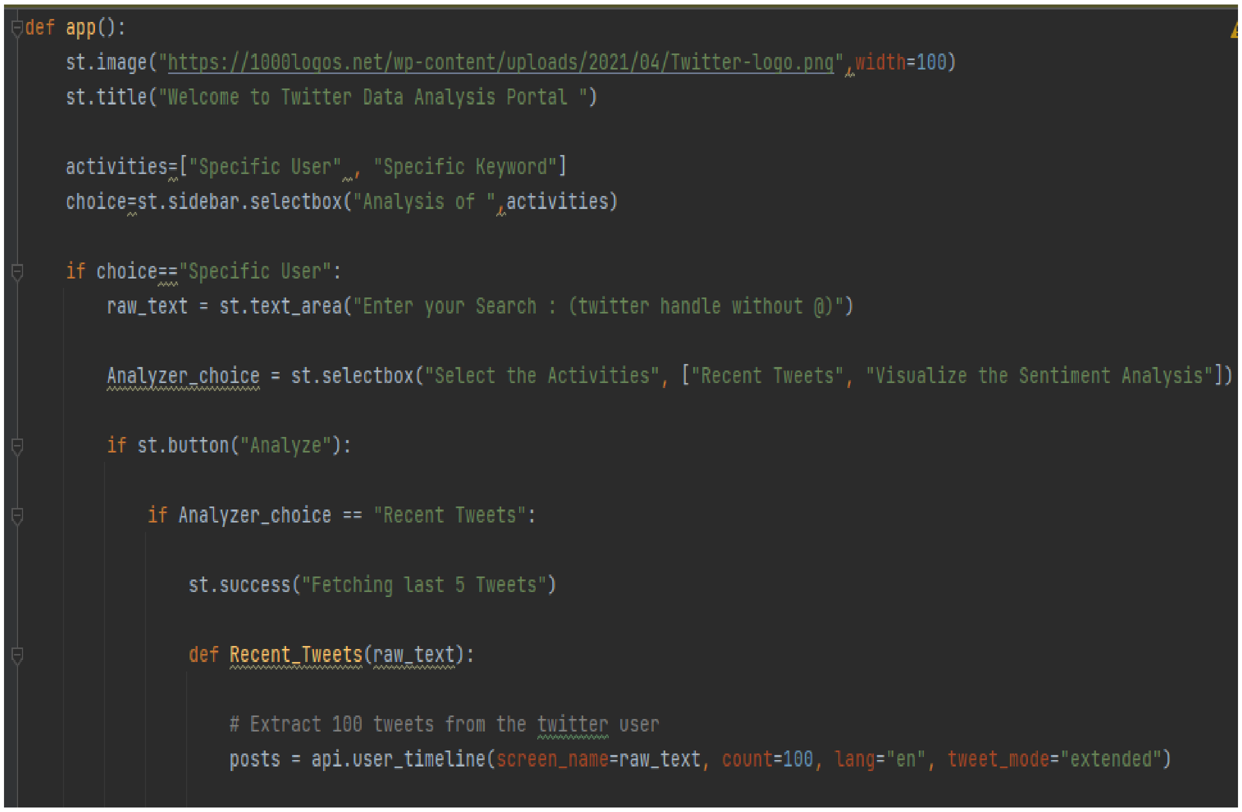
5.3.2) Creating Authentication



```
# Create the authentication object
authenticate = tweepy.OAuthHandler(consumerKey, consumerSecret)

# Set the access token and access token secret
authenticate.set_access_token(accessToken, accessTokenSecret)

# Creating the API object while passing in auth information
api = tweepy.API(authenticate, wait_on_rate_limit=True)


# plt.style.use('fivethirtyeight')
```

Fig 5.23: Authentication

5.3.3) Main function

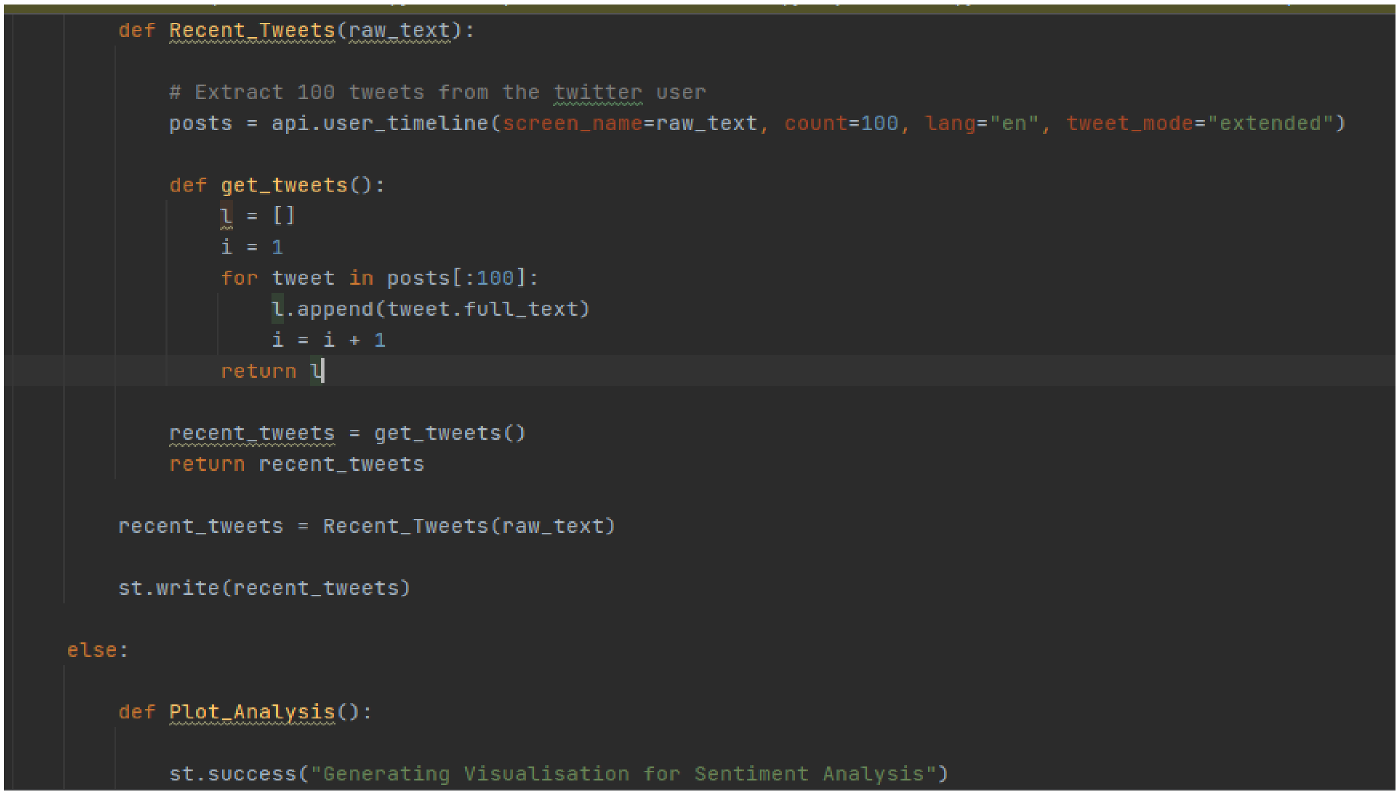


```
def app():
    st.image("https://1000logos.net/wp-content/uploads/2021/04/Twitter-logo.png",width=100)
    st.title("Welcome to Twitter Data Analysis Portal ")

    activities=["Specific User", "Specific Keyword"]
    choice=st.sidebar.selectbox("Analysis of ",activities)

    if choice=="Specific User":
        raw_text = st.text_area("Enter your Search : (twitter handle without @)")

        Analyzer_choice = st.selectbox("Select the Activities", ["Recent Tweets", "Visualize the Sentiment Analysis"])

        if st.button("Analyze"):

            if Analyzer_choice == "Recent Tweets":

                st.success("Fetching last 5 Tweets")

                def Recent_Tweets(raw_text):

                    # Extract 100 tweets from the twitter user
                    posts = api.user_timeline(screen_name=raw_text, count=100, lang="en", tweet_mode="extended")
```

Fig 5.24: Main function

We can perform activities for a Specific user or we can use keyword to **analyse.**

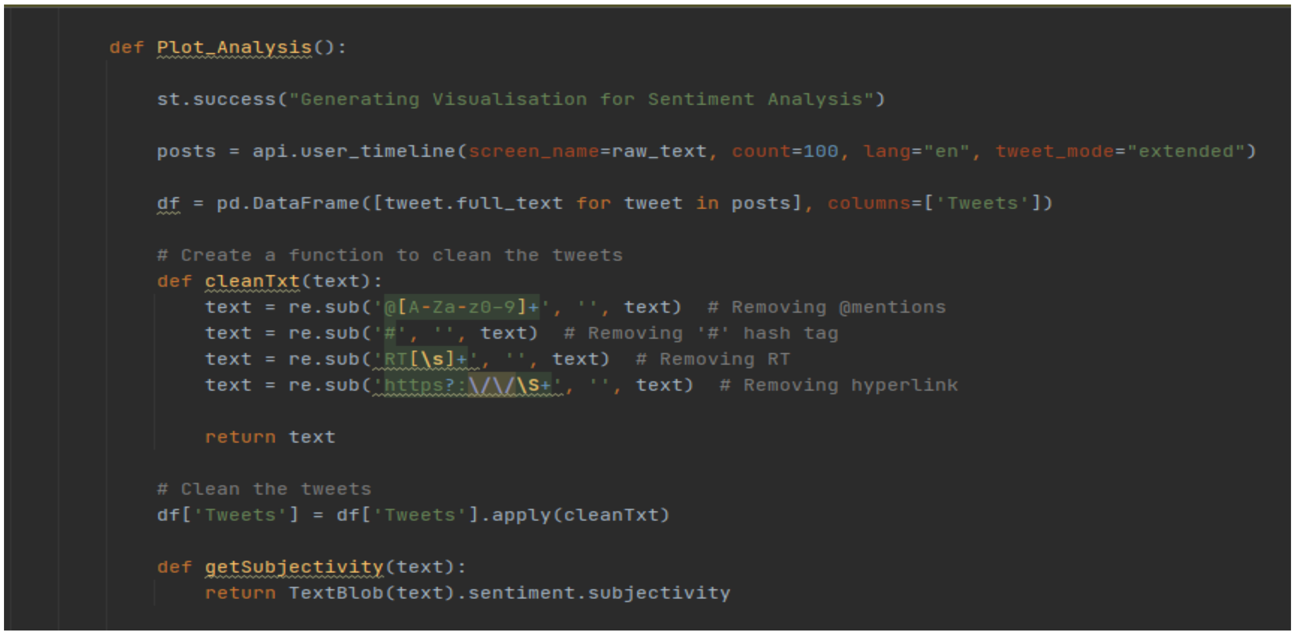
5.3.4) Extracting tweets from the user



```
        def Recent_Tweets(raw_text):

            # Extract 100 tweets from the twitter user
            posts = api.user_timeline(screen_name=raw_text, count=100, lang="en", tweet_mode="extended")

            def get_tweets():
                l = []
                i = 1
                for tweet in posts[:100]:
                    l.append(tweet.full_text)
                    i = i + 1
                return l

            recent_tweets = get_tweets()
            return recent_tweets

        recent_tweets = Recent_Tweets(raw_text)

        st.write(recent_tweets)

    else:

        def Plot_Analysis():

            st.success("Generating Visualisation for Sentiment Analysis")
```

Fig 5.25: Extracting Tweets

5.3.5) Generating visualization for sentiment analysis:

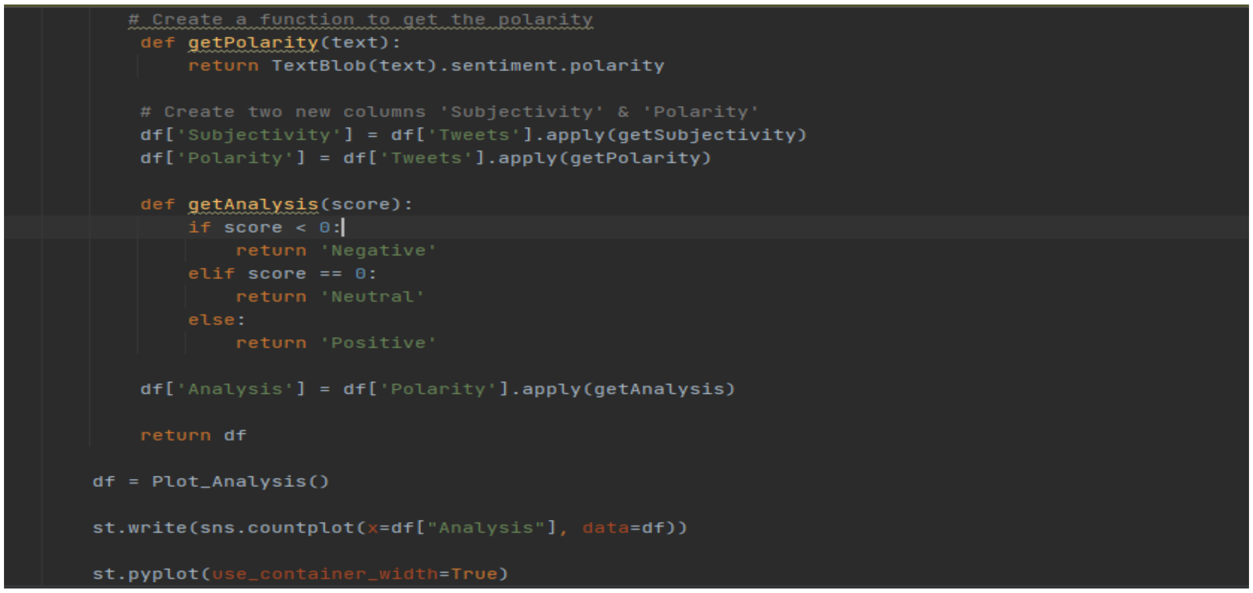


```python
def Plot_Analysis():

    st.success("Generating Visualisation for Sentiment Analysis")

    posts = api.user_timeline(screen_name=raw_text, count=100, lang="en", tweet_mode="extended")

    df = pd.DataFrame([tweet.full_text for tweet in posts], columns=['Tweets'])

    # Create a function to clean the tweets
    def cleanTxt(text):
        text = re.sub('@[A-Za-z0-9]+', '', text)  # Removing @mentions
        text = re.sub('#', '', text)  # Removing '#' hash tag
        text = re.sub('RT[\s]+', '', text)  # Removing RT
        text = re.sub('https?:\/\/\S+', '', text)  # Removing hyperlink

        return text

    # Clean the tweets
    df['Tweets'] = df['Tweets'].apply(cleanTxt)

    def getSubjectivity(text):
        return TextBlob(text).sentiment.subjectivity
```

Fig 5.26: Visualization

1. Here, the first function is Plot_Analysis(): which will generate visual representation of sentiment analysis using panda library.
2. After that cleanTxt function will remove mentions, hash tags, hyperlinks, Retweets from specific tweet which we will be analysisng and will give us clean text.

5.3.6) Creating function for getting the Polarity & Subjectivity

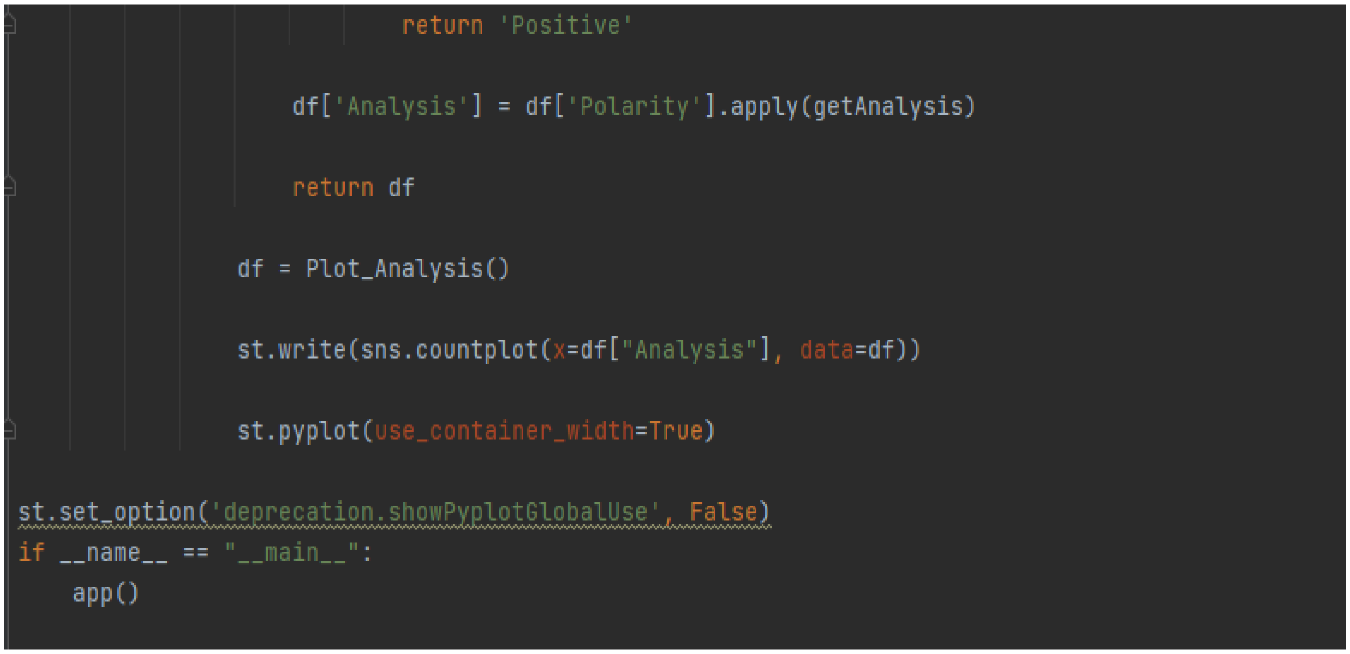


```python
    # Create a function to get the polarity
    def getPolarity(text):
        return TextBlob(text).sentiment.polarity

    # Create two new columns 'Subjectivity' & 'Polarity'
    df['Subjectivity'] = df['Tweets'].apply(getSubjectivity)
    df['Polarity'] = df['Tweets'].apply(getPolarity)

    def getAnalysis(score):
        if score < 0:
            return 'Negative'
        elif score == 0:
            return 'Neutral'
        else:
            return 'Positive'

    df['Analysis'] = df['Polarity'].apply(getAnalysis)

    return df

df = Plot_Analysis()

st.write(sns.countplot(x=df["Analysis"], data=df))

st.pyplot(use_container_width=True)
```

Fig 5.27 Polarity & Subjectivity

- Textblob→ sentiment analyser which calculates the polarity of sentences.
- Subjectivity→ Subjectivity is also a float which lies in the range of [0, 1].

5.3.7) Plotting analysis (output): -



```
                        return 'Positive'

                    df['Analysis'] = df['Polarity'].apply(getAnalysis)

                    return df

                df = Plot_Analysis()

                st.write(sns.countplot(x=df["Analysis"], data=df))

                st.pyplot(use_container_width=True)

st.set_option('deprecation.showPyplotGlobalUse', False)
if __name__ == "__main__":
    app()
```

Fig 5.28: Analysis

st.set_option('deprecation.showPyplotGlobalUse', False) is used to remove any warning coming on streamlit web app page.

5.4) TESTING APPROACH:

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

5.1TYPES OF TESTS

5.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic

tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.1.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:
Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.1.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.2) WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Kilometric complexity
- Deriving test cases
- Graph matrices Control

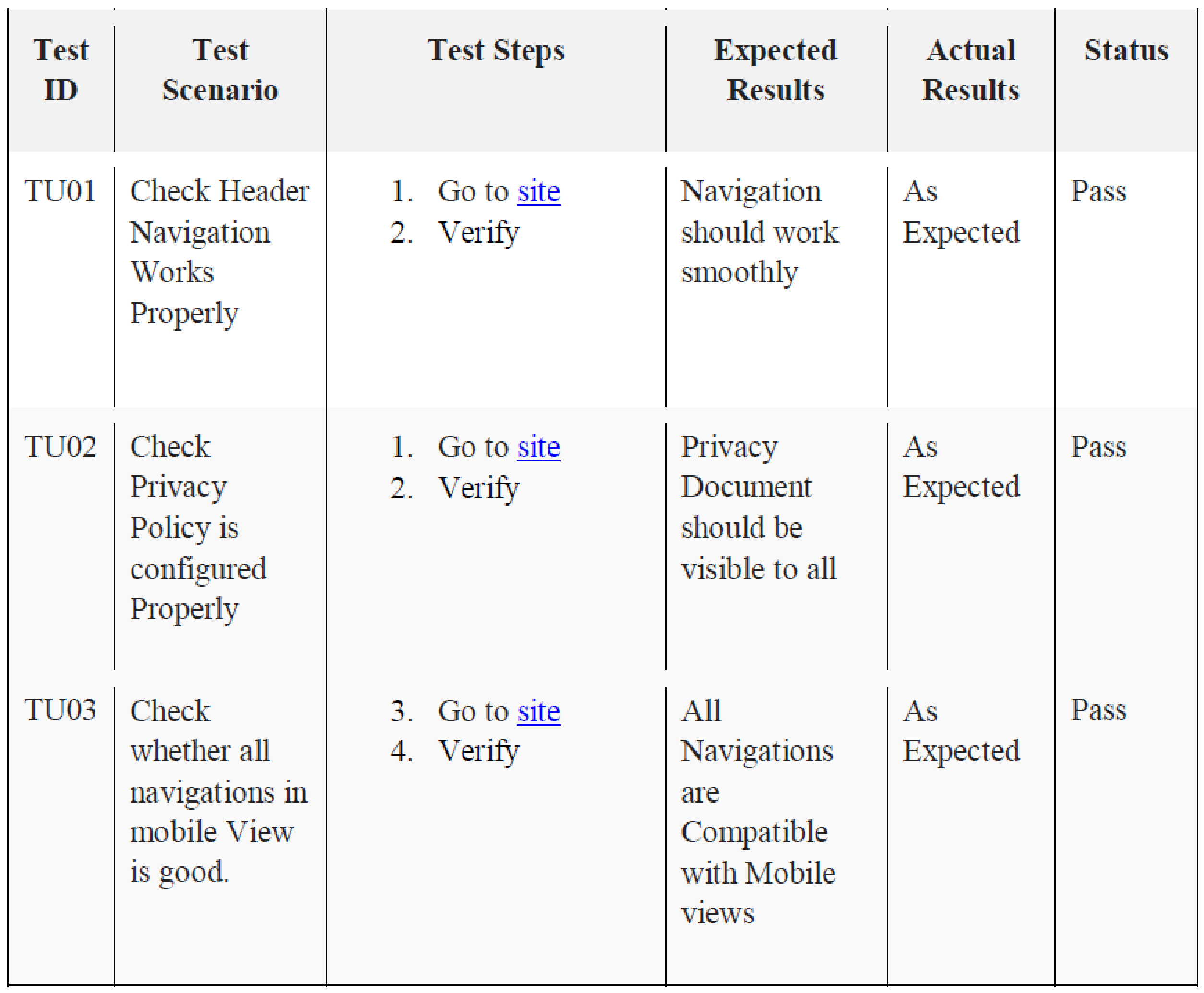
5.3) BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

5.4) INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures

caused by interface defects.
The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.



Test ID	Test Scenario	Test Steps	Expected Results	Actual Results	Status
TU01	Check Header Navigation Works Properly	1. Go to site 2. Verify	Navigation should work smoothly	As Expected	Pass
TU02	Check Privacy Policy is configured Properly	1. Go to site 2. Verify	Privacy Document should be visible to all	As Expected	Pass
TU03	Check whether all navigations in mobile View is good.	3. Go to site 4. Verify	All Navigations are Compatible with Mobile views	As Expected	Pass

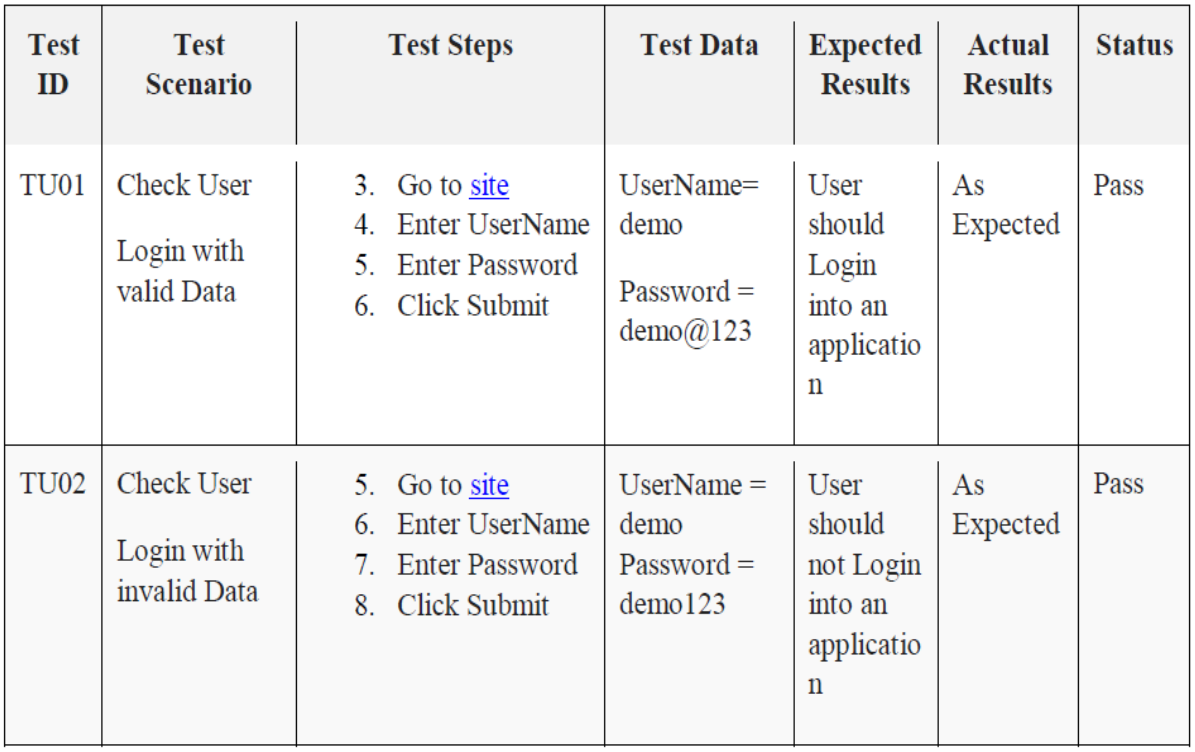
Test Results All the test cases mentioned above passed successfully. No defects encountered.

5.5) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

5.6) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.



Test ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Status
TU01	Check User Login with valid Data	3. Go to site 4. Enter UserName 5. Enter Password 6. Click Submit	UserName= demo Password = demo@123	User should Login into an application	As Expected	Pass
TU02	Check User Login with invalid Data	5. Go to site 6. Enter UserName 7. Enter Password 8. Click Submit	UserName = demo Password = demo123	User should not Login into an application	As Expected	Pass

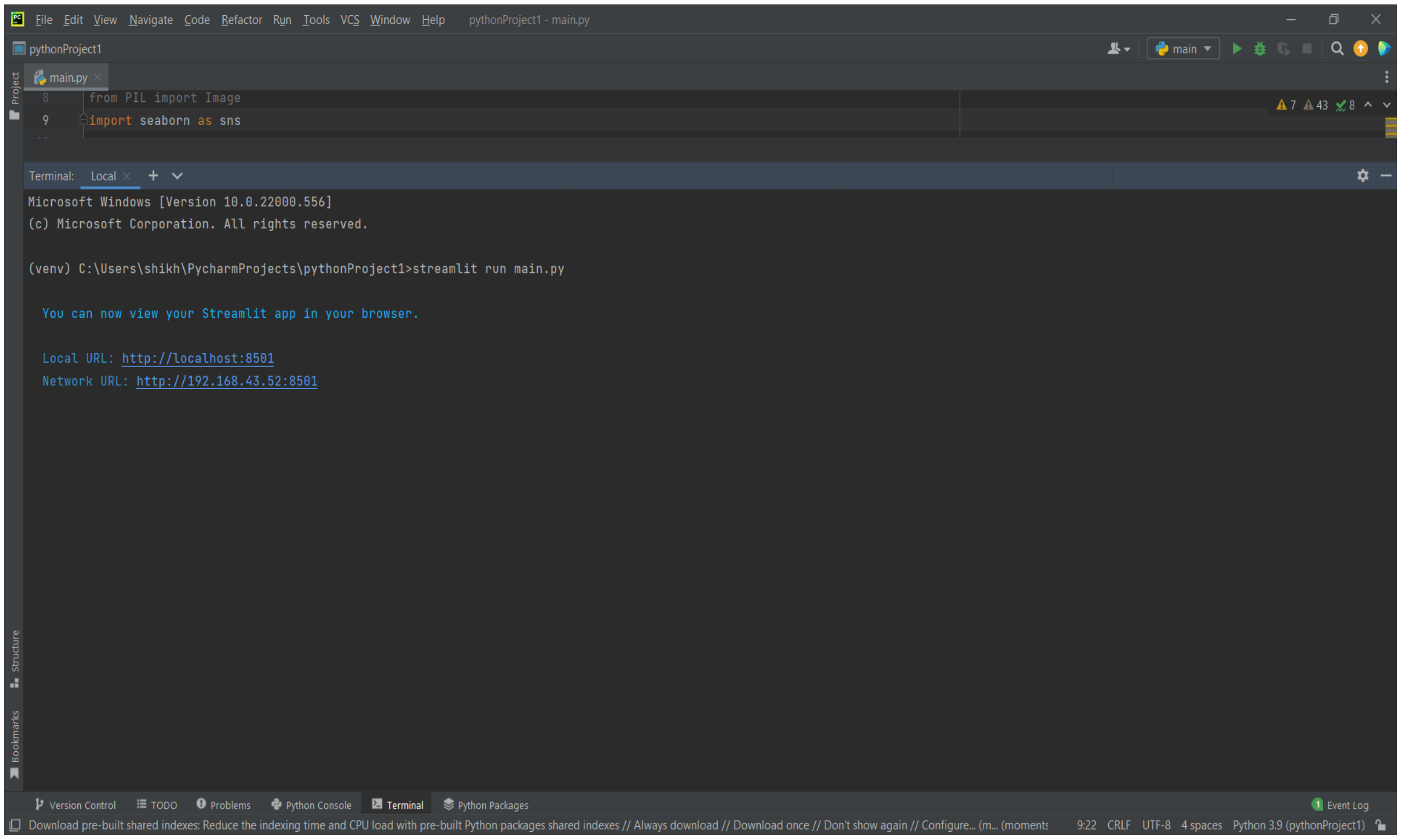
Test Results All the test cases mentioned above passed successfully. No defects encountered.

Chapter 6
Results and discussion

6.1) Result:

1) Open the pycharm editor and write the command in terminal

   Streamlit run main.py


File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject1 - main.py
pythonProject1
main.py
Terminal: Local
Version Control TODO Problems Python Console Terminal Python Packages Event Log

```
from PIL import Image
import seaborn as sns
```

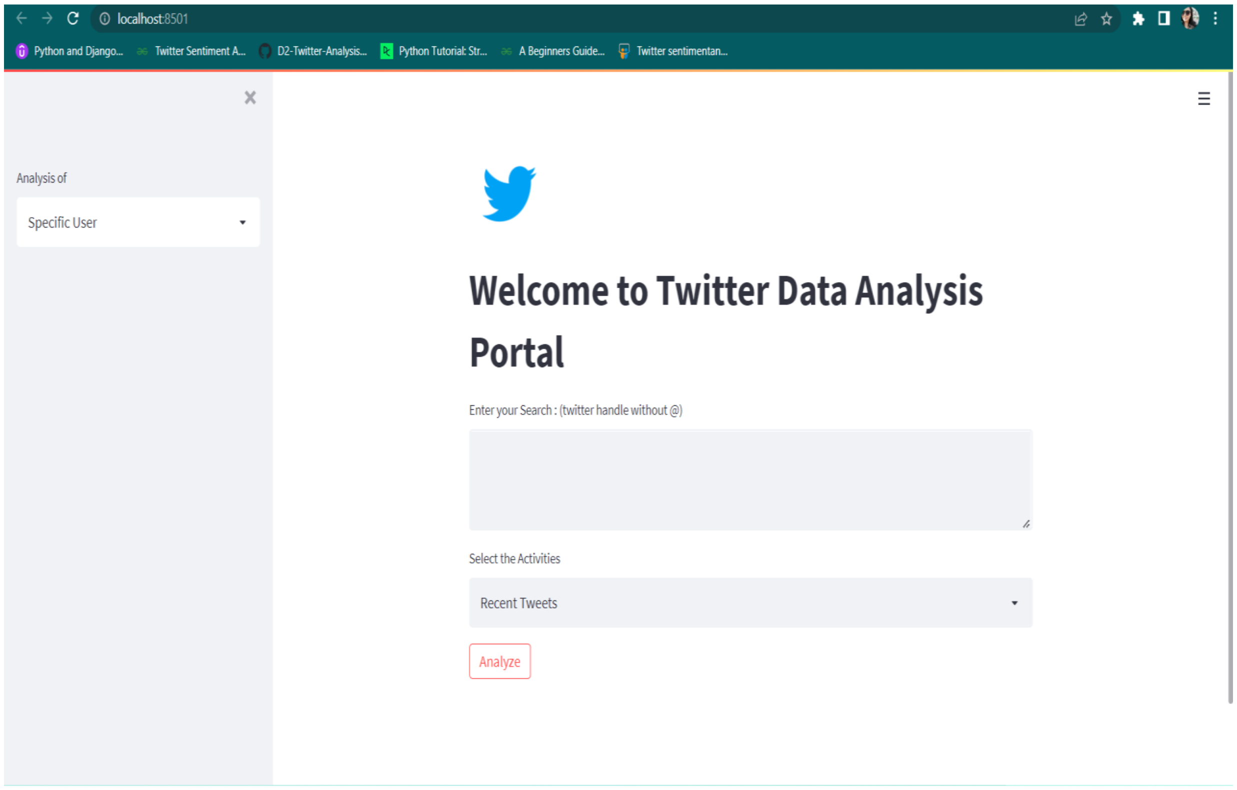
```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

(venv) C:\Users\shikh\PycharmProjects\pythonProject1>streamlit run main.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.43.52:8501
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (m... (moments 9:22 CRLF UTF-8 4 spaces Python 3.9 (pythonProject1)

Fig 6.1: Running main file

2) Project Main screen will appear :

- For Specific User:



Welcome to Twitter Data Analysis Portal

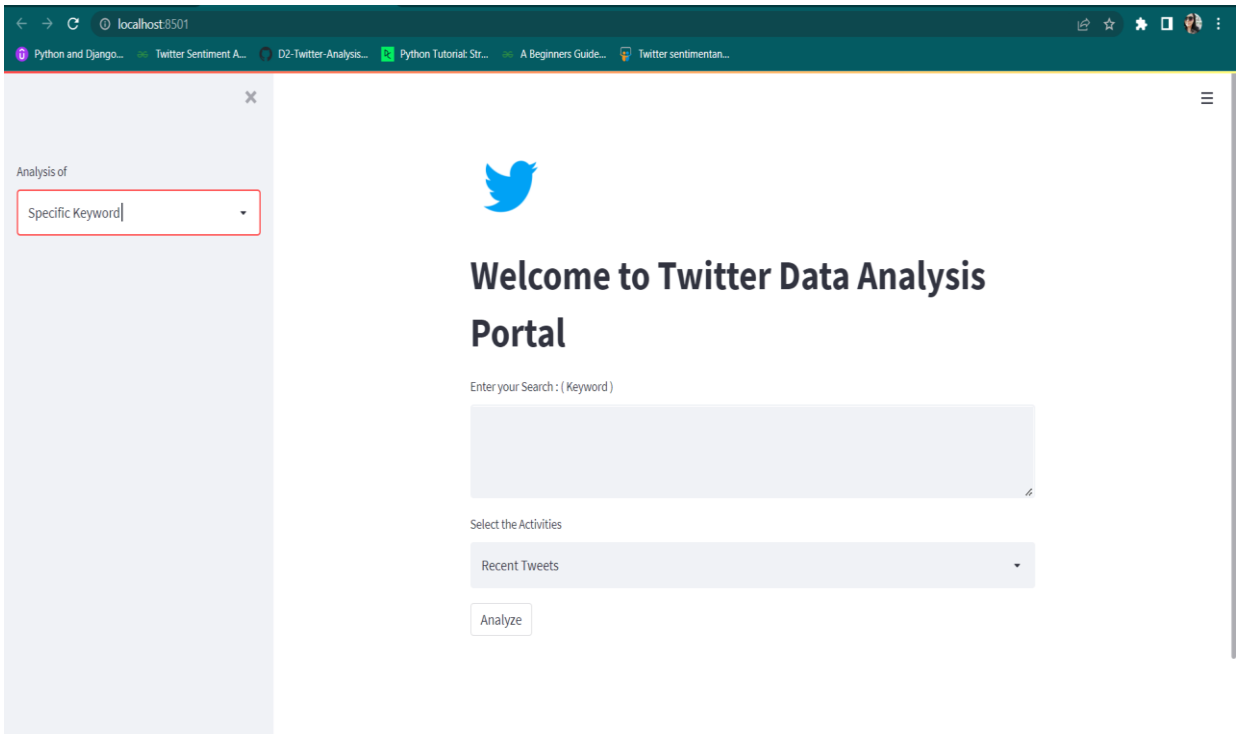
Enter your Search : (twitter handle without @)

Select the Activities

Recent Tweets

Fig 6.2 User Interface

- For Specific Keyword :



Analysis of

Specific Keyword

Welcome to Twitter Data Analysis Portal

Enter your Search : (Keyword)

Select the Activities

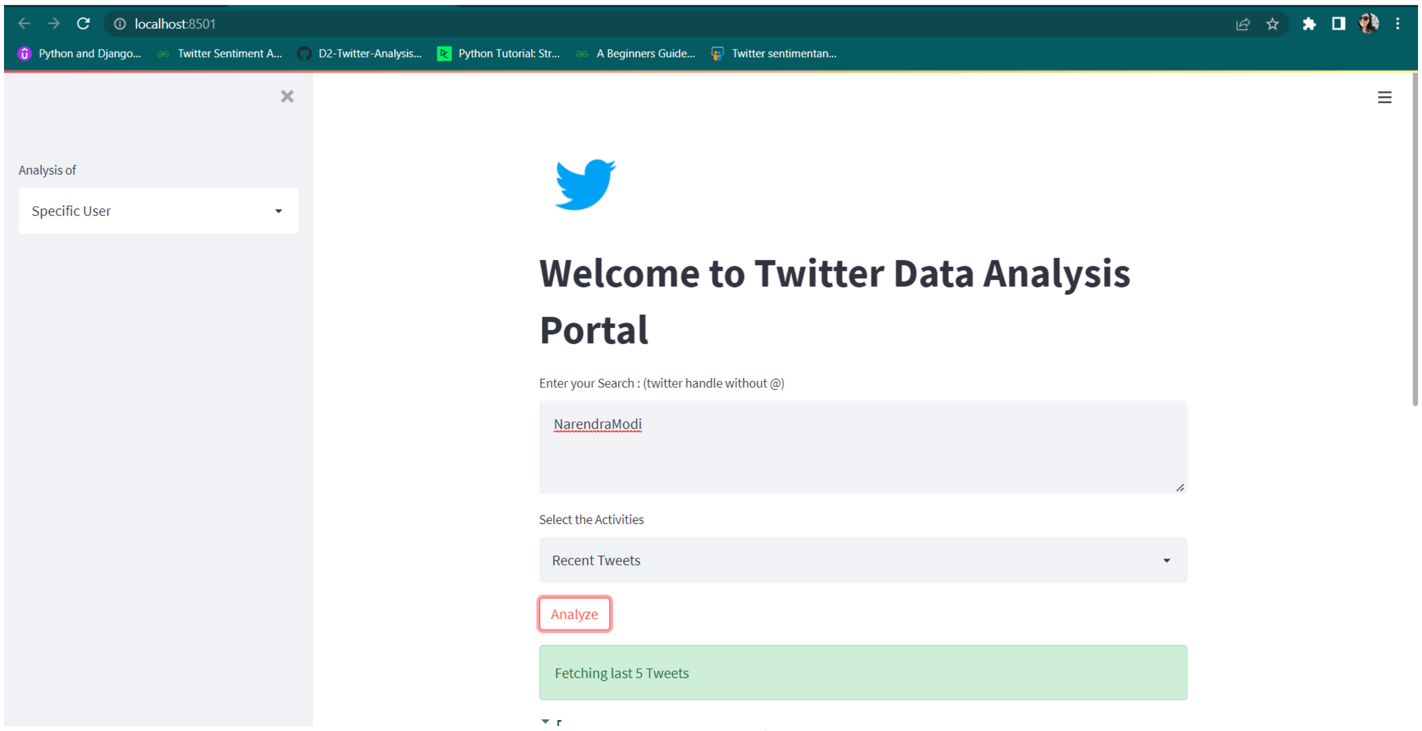
Recent Tweets

Analyze

Fig 6.3: User interface (For specific keyword)

3) After Entering our Search:

Our Search: NarendraModi Activity: Recent Tweets



Analysis of

Specific User

Welcome to Twitter Data Analysis Portal

Enter your Search : (twitter handle without @)

NarendraModi

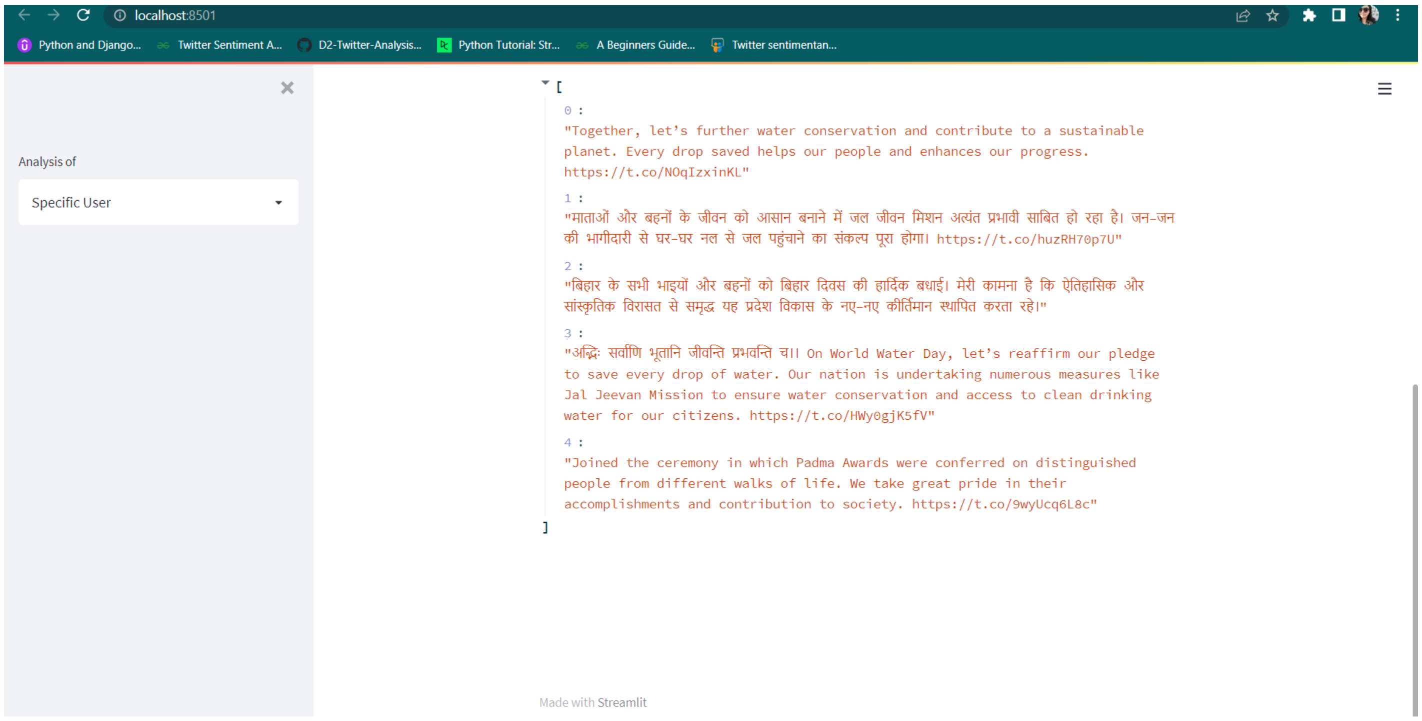
Select the Activities

Recent Tweets

Analyze

Fetching last 5 Tweets

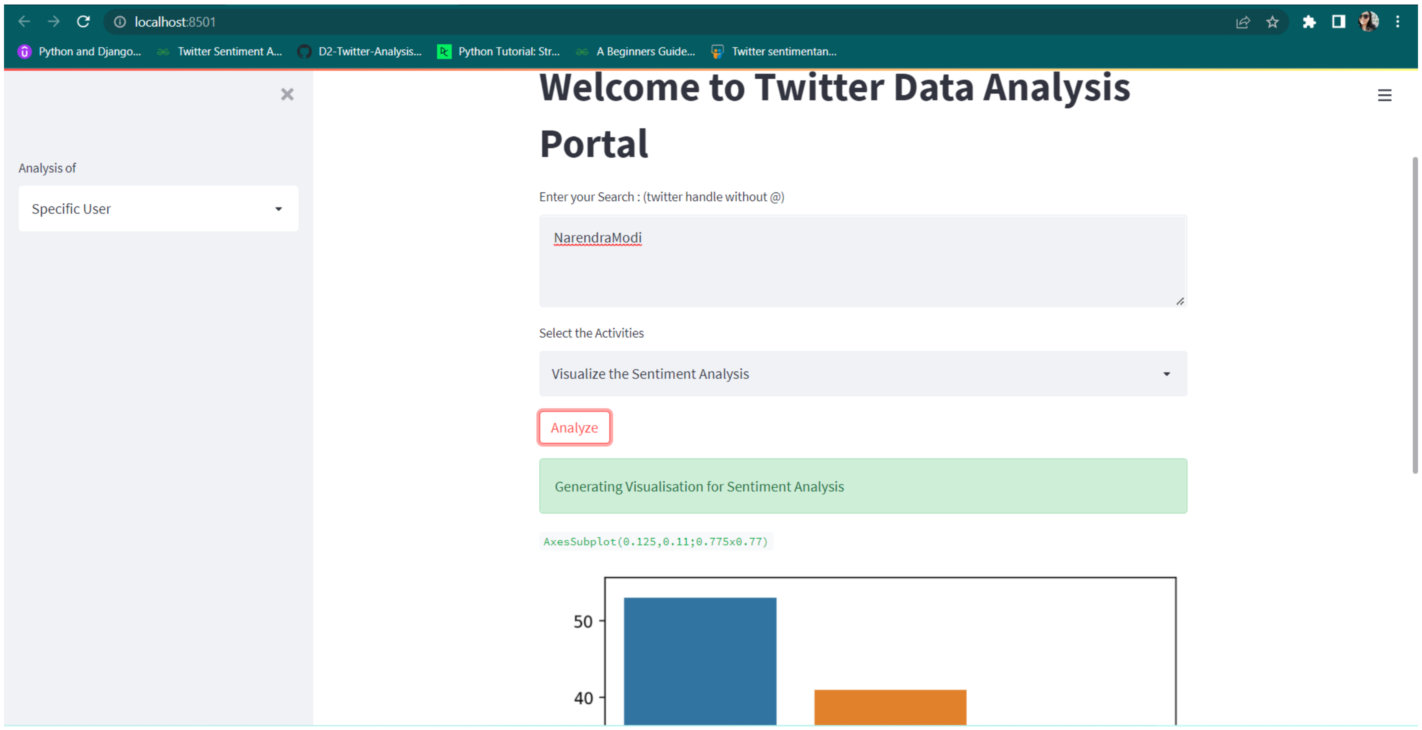
Fig 6.4(a): Recent tweets of a user



```
[
  0 :
  "Together, let’s further water conservation and contribute to a sustainable
  planet. Every drop saved helps our people and enhances our progress.
  https://t.co/NOqIzxinKL"
  1 :
  "माताओं और बहनों के जीवन को आसान बनाने में जल जीवन मिशन अत्यंत प्रभावी साबित हो रहा है। जन-जन
  की भागीदारी से घर-घर नल से जल पहुंचाने का संकल्प पूरा होगा। https://t.co/huzRH70p7U"
  2 :
  "बिहार के सभी भाइयों और बहनों को बिहार दिवस की हार्दिक बधाई। मेरी कामना है कि ऐतिहासिक और
  सांस्कृतिक विरासत से समृद्ध यह प्रदेश विकास के नए-नए कीर्तिमान स्थापित करता रहे।"
  3 :
  "अद्भिः सर्वाणि भूतानि जीवन्ति प्रभवन्ति च।। On World Water Day, let’s reaffirm our pledge
  to save every drop of water. Our nation is undertaking numerous measures like
  Jal Jeevan Mission to ensure water conservation and access to clean drinking
  water for our citizens. https://t.co/HWy0gjK5fV"
  4 :
  "Joined the ceremony in which Padma Awards were conferred on distinguished
  people from different walks of life. We take great pride in their
  accomplishments and contribution to society. https://t.co/9wyUcq6L8c"
]
```

Fig 6.4(b): Specific user

4) Visualizing sentiment Analysis on NarendraModi :



Analysis of

Specific User

Welcome to Twitter Data Analysis Portal

Enter your Search : (twitter handle without @)

NarendraModi

Select the Activities

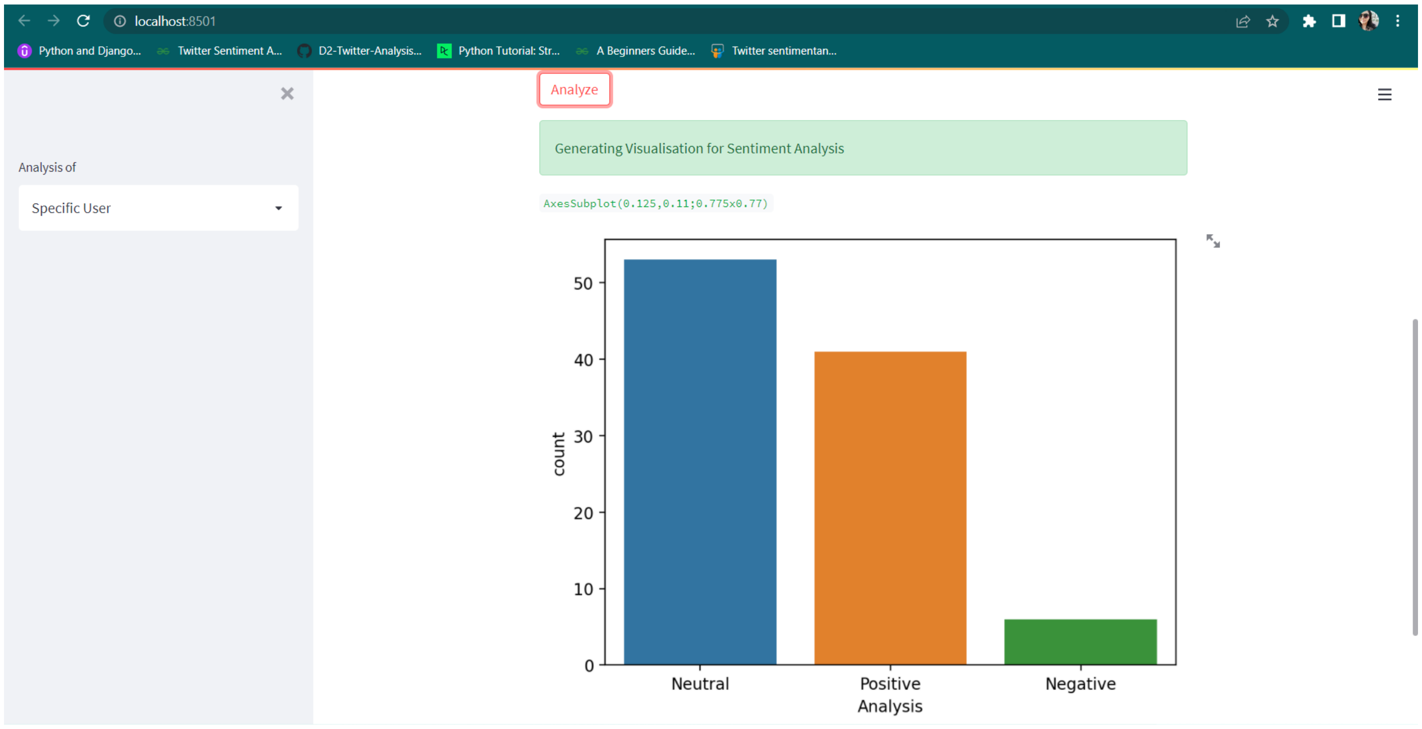
Visualize the Sentiment Analysis

Analyze

Generating Visualisation for Sentiment Analysis

```
AxesSubplot(0.125,0.11;0.775x0.77)
```

Fig 6.5: Sentiment analysis



Analysis of

Specific User

Analyze

Generating Visualisation for Sentiment Analysis

```
AxesSubplot(0.125,0.11;0.775x0.77)
```

Fig 6.6: Sentiment analysis

- For Specific Keyword :

  Search: Covid-19

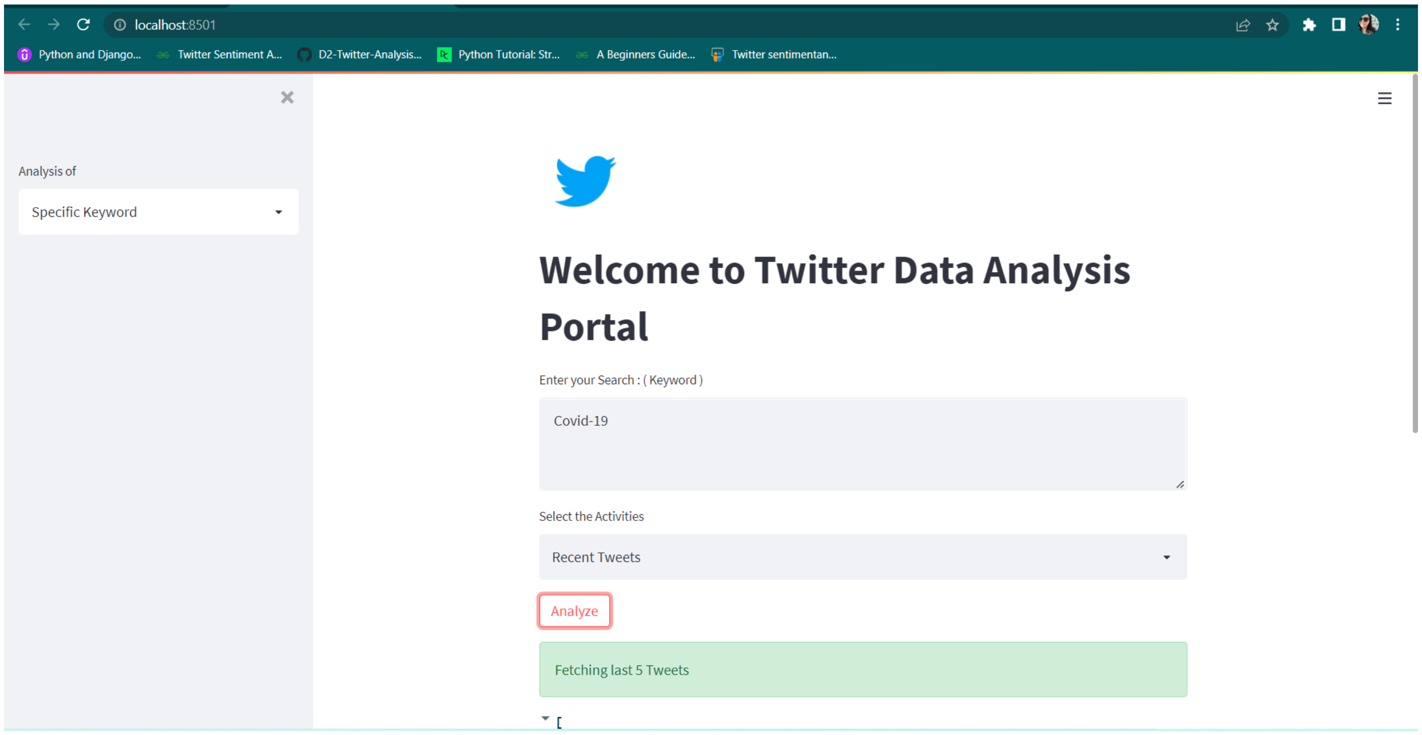
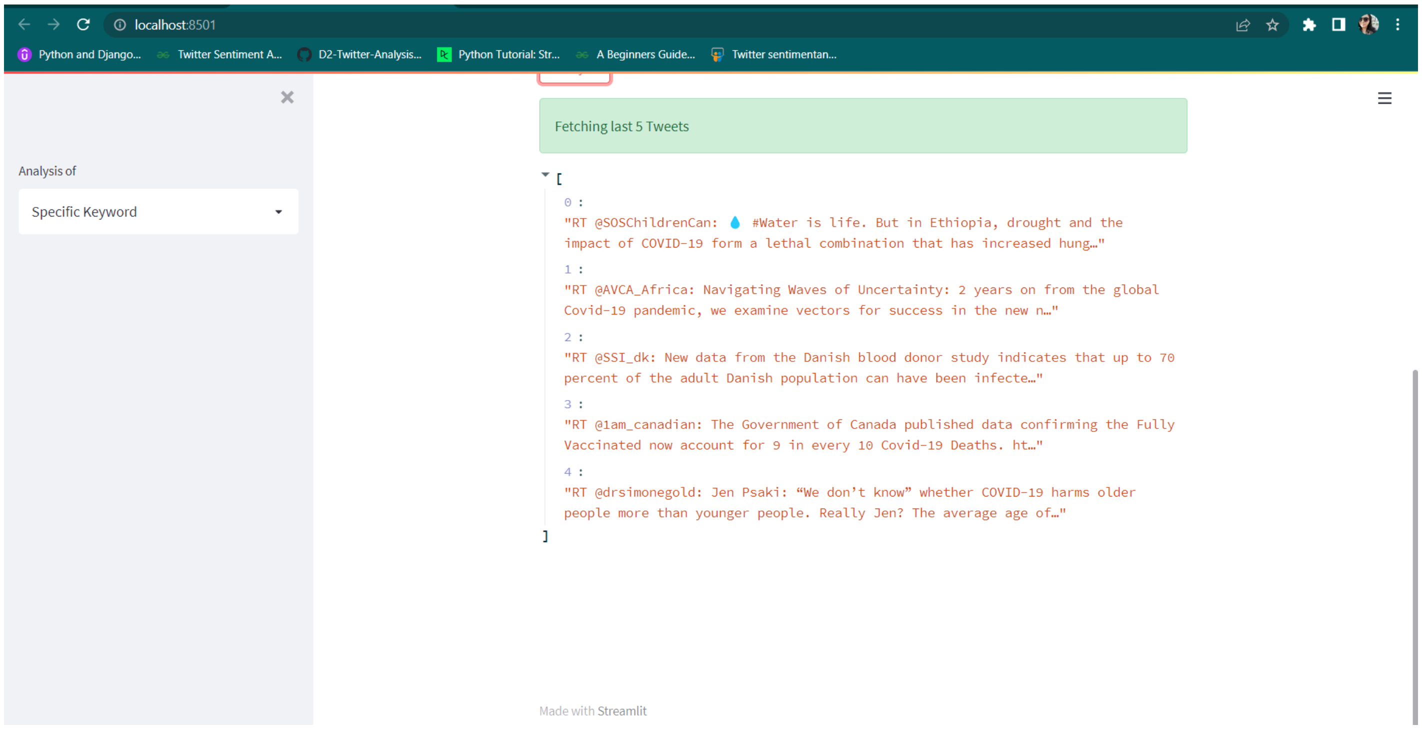


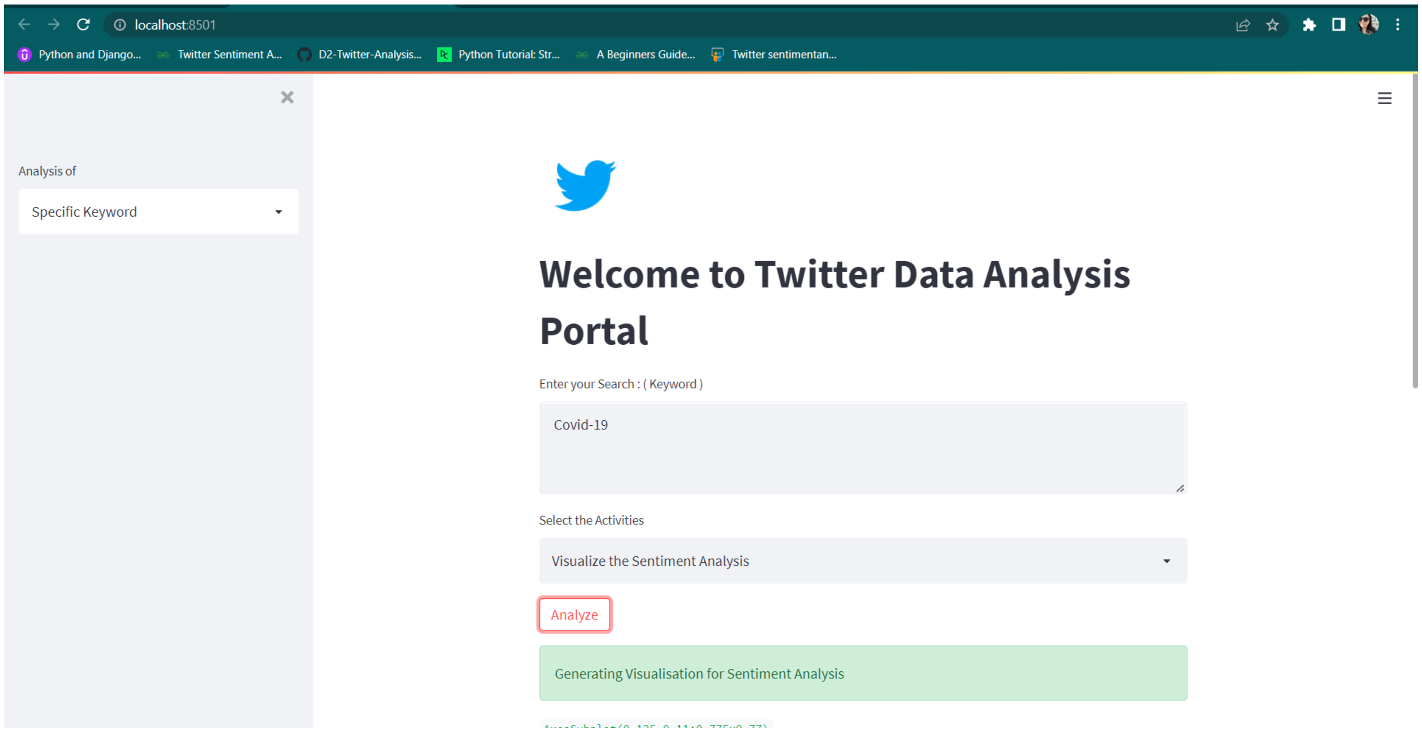
Fig 6.7: Specific Keyword



```
[
  0 :
  "RT @SOSChildrenCan: 💧 #Water is life. But in Ethiopia, drought and the
  impact of COVID-19 form a lethal combination that has increased hung…"
  1 :
  "RT @AVCA_Africa: Navigating Waves of Uncertainty: 2 years on from the global
  Covid-19 pandemic, we examine vectors for success in the new n…"
  2 :
  "RT @SSI_dk: New data from the Danish blood donor study indicates that up to 70
  percent of the adult Danish population can have been infecte…"
  3 :
  "RT @1am_canadian: The Government of Canada published data confirming the Fully
  Vaccinated now account for 9 in every 10 Covid-19 Deaths. ht…"
  4 :
  "RT @drsimonegold: Jen Psaki: “We don’t know” whether COVID-19 harms older
  people more than younger people. Really Jen? The average age of…"
]
```

Fig 6.8: Recent Tweets

5) Visualizing the sentiment analysis :


localhost:8501

Analysis of

Specific Keyword

Welcome to Twitter Data Analysis Portal

Enter your Search : (Keyword)

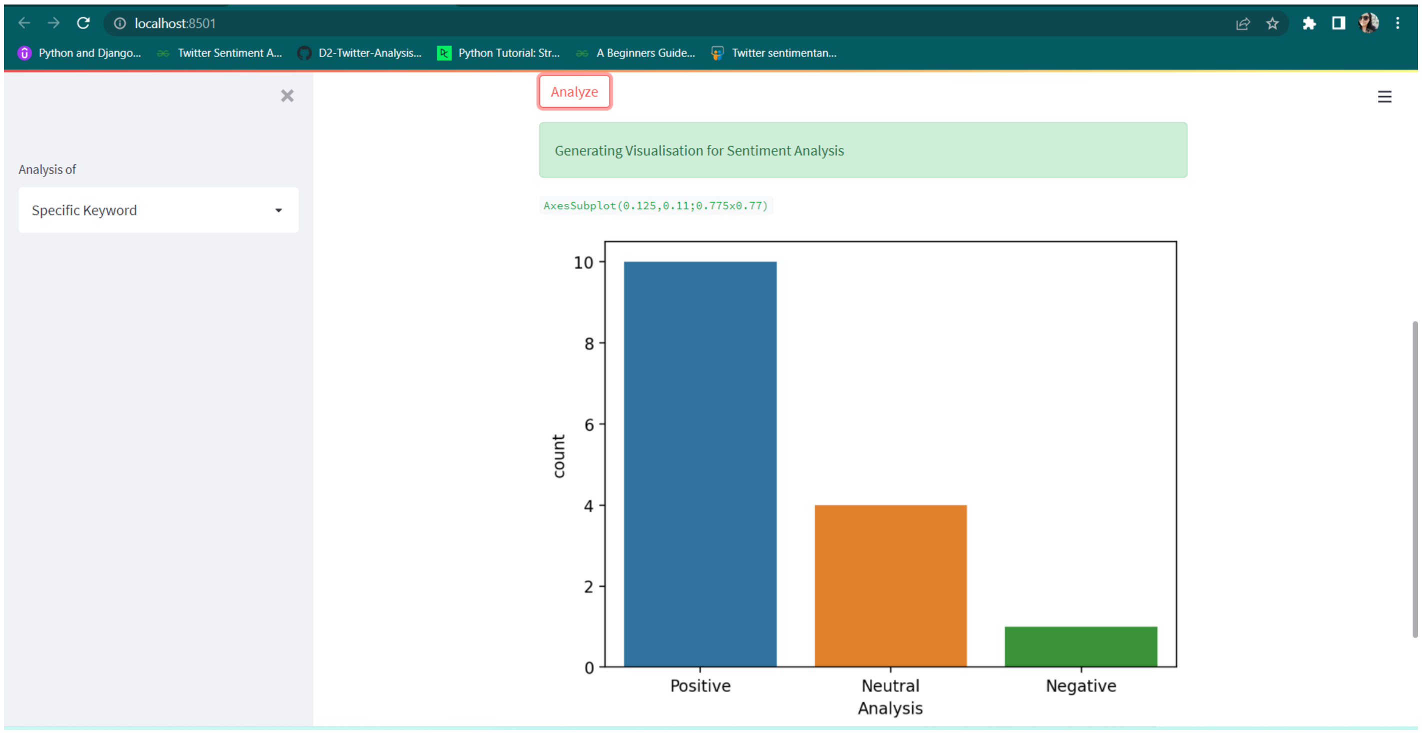
Covid-19

Select the Activities

Visualize the Sentiment Analysis

Generating Visualisation for Sentiment Analysis

Fig 6.9: Specific keyword


localhost:8501

Analysis of

Specific Keyword

Generating Visualisation for Sentiment Analysis

```
AxesSubplot(0.125,0.11;0.775x0.77)
```

Fig 6.10: Sentiment Analysis

Chapter 7

Conclusion and future work

7.1) Conclusion:

With the social networking era rapidly evolving, social media websites are a promising platformfor any individual to share their opinion, interests and ideas.

Reviews act as a valuable source of information for decision-making. Online e-commerce sites have provided their users to make their opinion about products and services. A huge amount of such opinions are publicly available in the form of reviews. Manufacturers, retailers as well as customers have great interest in customer reviews. A customer has an interest in such reviews todetermine which product or a particular brand to buy. Manufacturers can analyze the improvement area in their product from such opinionated reviews. Due to a large number of reviews available on the internet for analysis, it is not cost worthy to read these manually. To optimize this time-consuming task there is a need for an automated system that provides the summarized result of user sentiments.

So we built a live Twitter sentiment analyzer in a simple way. These kinds of apps are useful to businesses to know the sentiments of their brands on social media so that they can make informed decisions or at least address the issues that their customers have.

7.2) Future Work :

1) **Emotion Detection**

This sentiment analysis model detects the emotions that underlie a text. It makes associations between words and emotions like anger, happiness, frustration, etc. For example,

- Hubspot makes my day a lot easier :)' → Happiness
- 'Your customer service is a nightmare! Totally useless!!' → Anger

2) **Aspect-based Sentiment Analysis**

This type of sentiment analysis focuses on understanding the aspects or features that are being discussed in a given opinion. Product reviews, for example, are often composed of different opinions about different characteristics of a product, like Price, UX-UI, Integrations, Mobile Version, etc. Let’s see some examples:

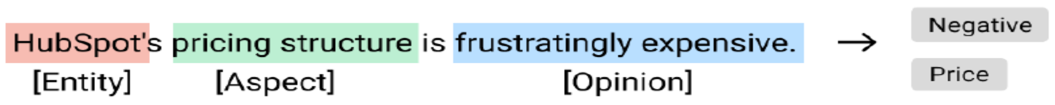


Fig 7.1: Example

3) Intent Detection

This type of sentiment analysis tries to find an action behind a given opinion, something that the user wants to do. Identifying user intents allows you to detect valuable opportunities to help customers, such as solving an issue, making improvements on a product or deriving complaints to the correspondent areas:

 “Very frustrated right now. Instagram keeps closing when I log in. Can you help?” → Request for Assistance

Customers experiencing issues can be easily spotted thanks to sentiment analysis.

REFERENCES

1. Sarlan, A., Nadam, C., & Basri, S. (2014, November). Twitter sentiment analysis. In *Proceedings of the 6th International conference on Information Technology and Multimedia* (pp. 212-216). IEEE.
2. Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)*, *15*, 2352.
3. Shelar, A., & Huang, C. Y. (2018, December). Sentiment analysis of twitter data. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 1301-1302). IEEE.
4. Bhavsar, H., & Manglani, R. (2019). Sentiment analysis of Twitter data using Python. *International Research Journal of Engineering and Technology (IRJET)*, *6*(3), 510-527.
5. Da Silva, N. F., Hruschka, E. R., & Hruschka Jr, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision support systems*, *66*, 170-179.
6. Gohil, S., Vuik, S., & Darzi, A. (2018). Sentiment analysis of health care tweets: review of the methods used. *JMIR public health and surveillance*, *4*(2), e5789.
7. https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/
8. https://monkeylearn.com/blog/sentiment-analysis-of-twitter/
9. https://towardsdatascience.com/sentiment-analysis-of-tweets-167d040f0583
10. https://www.youtube.com/watch?v=27P268Q7pE0
11. https://www.digitalvidya.com/blog/twitter-sentiment-analysis-introduction-and-techniques/
12. https://arxiv.org/ftp/arxiv/papers/1601/1601.06971.pdf
13. https://towardsdatascience.com/step-by-step-twitter-sentiment-analysis-in-python-d6f650ade58d
14. https://www.ibm.com/cloud/learn/natural-language-processing#:~:text=Natural%20language%20processing%20(NLP)%20refers,same%20way%20human%20beings%20can.
15. https://www.guru99.com/what-is-big-data.html#:~:text=Big%20Data%20is%20a%20collection,data%20but%20with%20huge%20size.
16. https://www.sas.com/en_in/insights/big-data/what-is-big-data.html
17. https://www.geeksforgeeks.org/libraries-in-python/
18. https://www.guru99.com/what-is-data-analysis.html
19. https://www.simplilearn.com/data-analysis-methods-process-types-article
20. https://www.datapine.com/blog/data-analysis-methods-and-techniques/
21. https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17
22. https://www.geeksforgeeks.org/what-is-sentiment-analysis/
23. https://towardsdatascience.com/how-to-use-twitters-api-c3e25c2ad8fe
24. https://blog.hubspot.com/website/how-to-use-twitter-api
25. https://www.jetbrains.com/pycharm/
26. https://www.jetbrains.com/pycharm/download/
27. https://en.wikipedia.org/wiki/PyCharm
28. https://www.python.org/
29. https://www.python.org/downloads/
30. https://www.w3schools.com/python/
31. https://www.tutorialspoint.com/pycharm/index.htm
32. https://www.jetbrains.com/help/pycharm/creating-and-running-your-first-python-project.html
33. https://docs.streamlit.io/

34. https://twitter.com/i/flow/login?input_flow_data=%7B%22requested_variant%22%3A%22eyJyZWRpcmVjdF9hZnRlcl9sb2dpbiI6Imh0dHBzOi8vZGV2ZWxvcGVyLnR3aXR0ZXIuY29tL2VuL3BvcnRhbC9wZXRpdGlvbi9lc3NlbnRpYWwvYmFzaWMtaW5mbyJ9%22%7D
35. https://developer.twitter.com/en/docs/twitter-api/getting-started/getting-access-to-the-twitter-api
36. https://www.analyticsvidhya.com/blog/2021/10/build-a-live-twitter-sentiment-analyzer-with-tweepy-huggingface-transformers-and-streamlit/
37. https://www.youtube.com/watch?v=eFdPGpny_hY
38. https://www.youtube.com/watch?v=fcWpAwgfw7o
39. https://www.youtube.com/watch?v=O_B7XLfx0ic