Project Report

on

**SENTOMIZATION**

Submitted

In Partial Fulfillment of

**MASTER OF COMPUTER APPLICATIONS (MCA)**

**Submitted by:**



Name – Arjun.V

(Roll No)-24/SCA/MCA/057

**Under the Supervision of:**

(Name of Project Guide, Designation)



**School of Computer Applications**

**Manav Rachna International Institute of Research and Studies**

**(DEEMED TO BE UNIVERSITY)**

Sector-43, Aravalli Hills

Faridabad – 121001

**June 2025**

**DECLARATION**

I do hereby declare that this project work entitled “**Sentomization**” submitted by me for the partial fulfilment of the requirement for the award of **MASTER OF COMPUTER APPLICATIONS** is a record of my own work. The report embodies the finding based on my study and observation and has not been submitted earlier for the award of any degree or diploma to any Institute or University

**SIGNATURE**

# CERTIFICATE

This is to certify that the project report entitled “**Sentomization**” submitted in partial fulfilment of the degree of **MASTER OF COMPUTER APPLICATIONS** to

Manav Rachna International Institute of Research and Studies, Faridabad is carried out by Mr./ Ms.\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Roll No), \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ under my guidance.

**Head of Department**

Name: Prof. (Dr.) Suhail Javed Quraishi

Date:

## 

# ACKNOWLEDGEMENT

I gratefully acknowledge for the assistance, cooperation, guidance and clarification provided by Ms./Mr.*\_\_\_\_\_\_\_\_\_\_\_\_\_* during the development of “**Sentomization**”. My extreme gratitude to \_*\_\_\_\_\_\_\_\_\_\_\_\_* who guided us throughout the project. Without his willing disposition, spirit accommodation, frankness, timely clarification and above all faith in us, this project could not have been completed in due time. Her readiness to discuss all important matters at work deserves special attention of. I would also like to thank all the faculty members of the computer application department for their cooperation and support.

I would like to extend my sincere gratitude to **Prof. (Dr.) Suhail Javed Quraishi – HOD** for his valuable teaching and advice. I would again like to thank all faculty members of the department for their cooperation and support. I would like to thank non-teaching staff of the department for their cooperation and support.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **TITLE** | **PAGE NO.** |
| **1** | **Declaration** |  |
| **2** | **Certificate** |  |
| **3** | **Acknowledgement** |  |
| **4** | **Abstract** |  |
| **5** | **Introduction** |  |
| **6** | **System Analysis Study** |  |
| **7** | **System Design** |  |
| **8** | **Software coding, testing and implementation** |  |
| **9** | **Output Forms and Report** |  |
| **10** | **Conclusion and Future Enhancement** |  |
| **11** | **Bibliography** |  |

## ABSTRACT

The field of text data analysis is an exciting new research direction due to a large number of real-world applications where discovering people’s opinions is important in better decision-making. The development of techniques for document-level text data analysis is one of the significant components of this area. Recently, people have started expressing their opinions on the Web which increased the need of analysing opinionated online content for various real-world applications. A lot of research is present in the literature for detecting text from the text. Still, there is a huge scope for improvement of these existing text data analysis models. Existing text data analysis models can be improved further with more semantic and common-sense knowledge.

### INTRODUCTION

Text data analysis, also known as text mining or text analytics, is the process of analysing and extracting useful information from large volumes of unstructured text data. This unstructured data can come from a variety of sources, such as social media, customer texts, emails, news articles, and more.

The goal of text data analysis is to transform this unstructured data into structured data that can be used for further analysis. This can involve several steps, such as text preprocessing (cleaning and preparing the text data), feature extraction (identifying key features of the text data), and modelling (applying statistical or machine learning algorithms to the data).

Text data analysis can be used in a variety of applications, such as text data analysis (determining the emotional tone of a piece of text), topic modelling (identifying key themes within the text data), and language translation. It is an important tool for businesses looking to gain insights into customer opinions and preferences, as well as for researchers studying trends and patterns in large volumes of text data.

#### **DATA CLEANING**

Remove duplicate or irrelevant observations

Remove unwanted observations from your dataset, including duplicate observations or irrelevant observations. Duplicate observations will happen most often during data collection. When you combine data sets from multiple places, scrape data, or receive data from clients or multiple departments, there are opportunities to create duplicate data.

Fix structural errors

Structural errors are when you measure or transfer data and notice strange naming conventions, typos, or incorrect capitalization. These inconsistencies can cause mislabeled categories or classes.

Filter unwanted outliers

Often, there will be one-off observations where, at a glance, they do not appear to fit within the data you are analyzing. If you have a legitimate reason to remove an outlier, like improper data entry, doing so will help the performance of the data you are working with.

Handle missing data

You can’t ignore missing data because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Neither is optimal, but both can be considered.

#### **NLP**

Natural Language Processing (NLP) is a branch of artificial intelligence dealing with the interaction between humans and computers using a natural language. The ultimate aim of NLP is to read, understand, and decode human words in a valuable manner. Most of the NLP techniques depend on machine learning to obtain meaning from human languages.

* **TEXT DATA**.

Text data is any data that is represented in the form of human-readable text, such as words, sentences, and paragraphs. This can include data from various sources such as social media, customer texts, emails, news articles, legal documents, scientific publications, and more.

Text data can be either structured or unstructured. Structured text data is organized and has a predefined format, such as tables or spreadsheets, while unstructured text data is free-form and does not have a specific structure. Unstructured text data is more difficult to analyse as it may contain spelling errors, grammatical errors, abbreviations, and different writing styles.

Text data is becoming increasingly important as more and more information is generated in digital form. Analysing this text data can help us gain insights into human behaviour, opinions, and preferences, and can be used in various applications such as text data analysis, topic modelling, and language translation.

#### **PURPOSE**

The purpose of text data analysis is to extract meaningful insights and knowledge from textual data. It involves applying various computational techniques and algorithms to process, interpret, and understand the content and structure of text data.

Text data analysis serves several purposes, including:

* **Information extraction:** By analyzing text data, valuable information can be extracted, such as key entities, events, relationships, sentiments, and opinions. This information can be used for various applications, such as customer feedback analysis, market research, and social media monitoring.

Text classification and categorization: Text data analysis enables the categorization and classification of text documents into specific topics or classes. This can be useful for organizing large volumes of text data, automating document sorting, and building recommendation systems.

* **Sentiment analysis:** Sentiment analysis involves determining the emotional tone or sentiment expressed in a piece of text. It can be used to understand customer opinions, gauge public sentiment towards a particular text or topic, and assist in reputation management.

Topic modelling: Text data analysis allows the identification of latent topics or themes within a collection of documents. This can help in exploring large text corpora, summarizing content, and uncovering trends or patterns in the data.

* **Text summarization:** Text data analysis techniques can be used to generate concise summaries of large documents or sets of documents. This is particularly useful for news articles, research papers, and legal documents, where quick access to key information is essential.
* **Natural language processing (NLP) applications:** Text data analysis is fundamental to many NLP applications, such as machine translation, question answering systems, chatbots, and voice assistants. It enables computers to understand, interpret, and generate human language.

Overall, text data analysis plays a crucial role in transforming unstructured textual data into structured, actionable insights, which can be leveraged for decision-making, automation, and improved understanding of human communication.

1. **SCOPE**

The scope of text data analysis is broad and diverse, as textual data is generated and consumed in virtually every field and industry. Here are some examples of the scope of text data analysis:

Business: Text data analysis is widely used in the business world to gain insights from customer feedback, social media monitoring, market research, and sales data. It can help businesses make informed decisions, improve customer engagement, and identify new market opportunities.

Healthcare: Text data analysis can be used to analyze electronic medical records, clinical notes, and research papers to improve patient care, identify potential drug interactions, and detect disease outbreaks.

Education: Text data analysis can help educators analyze student performance, detect plagiarism, and assess the effectiveness of teaching methods.

Legal: Text data analysis can be used to search legal documents for relevant information, analyze contracts, and identify patterns in legal language.

Government: Text data analysis can help government agencies analyze public opinion, monitor social media, and detect potential security threats.

Social sciences: Text data analysis can be used to analyze social media data, surveys, and interviews to gain insights into human behavior, opinions, and attitudes.

Media and entertainment: Text data analysis can help media and entertainment companies analyze viewer feedback, understand audience preferences, and develop personalized content recommendations.

The scope of text data analysis is continually expanding as new sources of textual data emerge, and more sophisticated computational techniques are developed to analyze and extract insights from this data. As a result, text data analysis is becoming increasingly important in many fields and industries, and it is expected to continue to grow in importance in the future.

#### **OBJECTIVE**

The Internet is a rich source of texts on e-commerce texts or online services. The customer always prefers to read texts before paying money to the service provider. But it is hardly possible to read all texts in today's fast life. Also, every text may provide new information of text or features of the text. So there is the probability of missing any important text given the by the consumer.

We need to identify the polarity of the text i.e. whether it is positive, negative, or neutral. Text data analysis will assist us to find out the polarity of texts. Due to a large number of texts, visuals can be 80% more efficient than textual format. So if we visualize all the texts, it will make easier decision-making process for the consumes will be able to see all texts at a glance and he/she will take decisions faster.

Thus, our main objectives are:

1. Dealing with neutral texts: Output must consider neutral texts as they make an impact on decision-making.
2. Improved Efficiency: Many texts are given for single text. Using the MapReduce environment we can improve the efficiency of text data analysis.
3. Text data analysis: To determine the attitude of mass people towards a particular text or service.
4. User-oriented Data Visualization: Customers are mainly nontechnical

persons. So we aim to visualize results in a user-readable format.

#### **SDLC Methodologies**

As we know that SDLC stands for Software Development Life Cycle models, and these are a variety of processes of design, development, and testing that are used in the industry today. While there’s no best or standout SDLC methodology, it’s essential to be across the most common models that can be applied to projects within an organization.

In this project, we used the waterfall methodology which is one of the oldest surviving SDLC methodologies. It follows a straightforward approach: the project development team completes one phase at a time, and each phase uses information from the last one to move forward.

While this methodology does make the needs and outcomes clear, and gives each stage a well-defined starting and ending point, there are downsides to Waterfall’s rigidity. In fact, some experts believe the Waterfall model was never meant to be a working SDLC methodology for developing software because of how fixed it is in nature. Because of this, SDLC Waterfall methods are best used for extremely predictable projects.

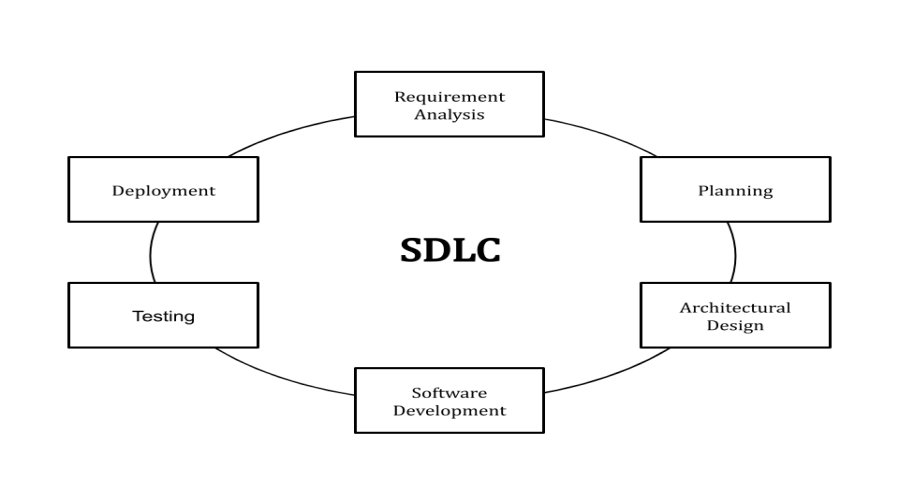


Fig: - 1.1

In this, a sequential phase order is followed. Here the phases do not overlap with each other.

The phases which we followed to develop this project are given below:

**Feasibility Study**:

The feasibility study involves understanding the problem and then determining the various possible strategies to solve the problem. These different identified solutions are analyzed based on their benefits and drawbacks, The best solution is chosen and all the other phases are carried out as per this solution strategy.

**Requirement gathering and analysis:**

Firstly, all the requirements regarding the software are gathered from the customer and then the gathered requirements are analyzed. The goal of the analysis part is to remove incompleteness (an incomplete requirement is one in which some parts of the actual requirements have been omitted) and inconsistencies (an inconsistent requirement is one in which some part of the requirement contradicts some other part).

**Design**:

The goal of this phase is to convert the requirements acquired in the SRS into a format that can be coded in a programming language. It includes high-level and detailed design as well as the overall software architecture. A Software Design Document is used to document all of this effort (SDD)

**Coding and Unit testing**:

In the coding phase software design is translated into source code using any suitable programming language. Thus, each designed module is coded. The aim of the unit testing phase is to check whether each module is working properly or not.

**Integration and System testing**:

Integration of different modules is undertaken soon after they have been coded and unit tested. Integration of various modules is carried out incrementally over a number of steps. During each integration step, previously planned modules are added to the partially integrated system and the resultant system is tested. Finally, after all the modules have been successfully integrated and tested, the full working system is obtained and system testing is carried out on this.

#### **SYSTEM ANALYSIS STUDY**

##### **INTRODUCTION**

Text data analysis is a branch of artificial intelligence that focuses on understanding and analyzing the emotional context of written texts. It is used to identify and measure the attitude, opinions, and emotions expressed in a text. It can be used to understand customer feedback, detect public opinion about a brand or text, or even monitor social media conversations. The aim of text data analysis is to provide insight into how people feel about certain topics or texts by analyzing the words they use in their conversations online. This helps businesses gain valuable insights into their customers’ feelings and preferences which can then be used to improve customer experience and increase sales.

The aim is to understand the overall opinion of the public about a particular topic or text. It helps companies gain insights into customer behavior and preferences by tracking public opinion over time. By using text data analysis, companies can better understand their customers’ needs and wants and use that information to improve their texts or services.

Natural Language Processing is a subfield of artificial intelligence that deals with the interaction between computers and human languages. It helps machines understand, interpret, and manipulate human language in order to generate meaningful insights from textual data. NLP technologies are used in a wide range of applications such as automated customer service, text data analysis, text summarization, and machine translation. With the help of NLP technologies, machines can process large amounts of unstructured text data to gain valuable insights and make decisions faster than ever before.

###### **PROJECT OVERVIEW**

The TEXT DATA ANALYSIS SYSTEM is to create an online-Information about the text that is related to the texts of any text. Through this software, any person who is interested in reading or giving texts can register himself. This project examines how a text expresses emotion. Customer feedback, survey replies, and text texts are all frequent uses. This can be useful in various situations, including social media monitoring, reputation management, and customer service.

###### **1.2 PROBLEM FOUND IN EXISTING SYSTEM**

* At present there is no software to keep any records of the texts.
* It becomes difficult to provide any record immediately.
* Privacy is difficult.
* Time-consuming is retiring the sentiments.
* It needs up gradation.

##### **SYSTEM REQUIREMENTS GATHERING**

Requirement Gathering or commonly known as the Discovery Phase is basically a process in which we understand and identify a business’s project technical requirements and proceed with a well-defined plan.

Although the discovery phase is an essential phase in any critical project plan, it is quite often overlooked in the absence of sufficient groundwork.

Some of the project consultants might argue that if a client’s requirements are accurately identified in the early stages, then completing a full phase of requirements gathering is simply not needed. One can actually not stress enough the importance of this recovery phase.

If the requirement gathering is not done properly, it can result in project deliverables not meeting the business requirements which in turn would result in a waste of time and money.

This phase is critical as the information gathered will be utilized as a base for the System Requirements Specifications (SRS) document.

The SRS will include a vision or mission statement of the client company defining the overall objectives and business plans.

So basically, the requirements-gathering phase enables both parties to minimize risks and balance the task management within the required timeframe. A process of discovery phase must always include key executives and stakeholders.

###### 2.3. FEASIBILITY STUDY

A preliminary investigation examines project feasibility and the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility of adding new modules and debugging the old running system. All system is feasible if they are unlimited resources and infinite time.

There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility
* Economic Feasibility

2.3.1 TECHNICAL FEASIBILITY

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation

System’. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to the users.

The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides a technical guarantee of accuracy, reliability, and security.

The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the system.

2.3.2 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost of creating the system is evaluated against the ultimate benefit derived from the new systems.

Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any additional hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

2.3.3 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information systems.

That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised to test the operational feasibility of a project include the following: -

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.3.4. SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements:

1. Processor: Minimum 1 GHz
2. Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
3. Hard Drive: Minimum 32 GB 4. Memory (RAM): Minimum 1 GB

Software Requirements:

1. Browser: All browsers are supported
2. Operating System: All operating systems are supported
3. Microsoft Office
4. Anaconda3
5. Jupyter Notebook

**Chapter-3**

#### **SYSTEMS DESIGN**

##### 3.1 INTRODUCTION

System design is the process of defining elements of a system like modules, architecture, components, and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing, and designing systems that satisfy the specific needs and requirements of a business or organization.

Design methods:

1. Architectural design:

To describe the views, models, behavior, and structure of the system.

1. Logical design: To represent the data flow, inputs, and outputs of the system.
2. Physical design:
3. How users add information to the system and how the system represents information back to the user.
4. How the data is modeled and stored within the system.
5. How data moves through the system, how data is validated, secured, and/or transformed as it flows through and out of the system.

System design takes the following inputs −

Statement of work

Requirement determination plan

Current situation analysis

Proposed system requirements include a conceptual data model, modified DFDs, and Metadata (data about data).

3.2 PHYSICAL DESIGN:

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.

It consists of the following steps −

Specifying the input/output media, designing the database, and specifying backup procedures.

Planning system implementation.

Devising a test and implementation plan, and specifying any new hardware and software.

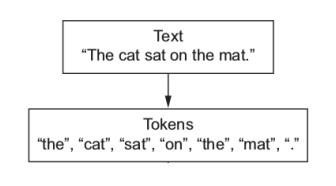
Updating costs, benefits, conversion rates, and system constraints

EXTRACTION OF DATA

**Tokenization:** Tokenization is the process of converting text into tokens before transforming it into vectors. It is also easier to filter out unnecessary tokens. For example, a document into paragraphs or sentences into words. In this case, we are tokenizing the texts into words.

**Removing punctuations and special symbols:**

Apart from the considered set of emoticons punctuations and symbols like &,\,;



are removed.

**Stop words removal**:

Stop words are the most commonly occurring words that are not relevant in the context of the data and do not contribute any deeper meaning to the phrase. In this case, contain no sentiment. NLTK provides a library used for this.

"This is a sample sentence, showing off the stop words filtration."

['This', 'is', 'a', 'sample', 'sentence', ',', 'showing', 'off', 'the', 'stop', 'words', 'filtration', '.']

**After stop words removal:**

'This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtration', '.']

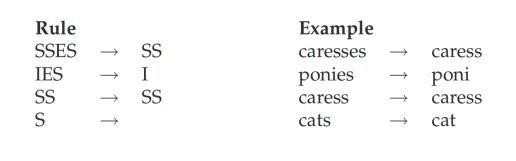
**Stemming and Lemmatization**:

Sentences are always narrated in tenses, singular and plural forms making most words accompanied with -ing,-ed,es and ies. Therefore, extracting the root word will suffice to identify text behind the text.

Base forms are the skeleton for grammar stemming and lemmatization reduces inflectional forms and derivational forms to common base forms.

Example: Cats is reduced to the cat, and ponies is reduced to the pony.

Stemming is a crude way of reducing terms to their root, by just defining rules of chopping off some characters at the end of the word, and hopefully, gets good results most of the time. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. With that being said, stemming/lemmatizing helps us reduce the number of overall terms to certain “root” terms.



**Feature Extraction:**

Text data demands a special measure before you train the model.

Words after tokenization are encoded as integers or floating point values for feeding input to the machine learning algorithm. This practice is described as vectorization or feature extraction. The scikit-learn library offers a TF-IDF vectorizer to convert text to word frequency vectors.

**Fitting Data to Classifier and predicting test data:**

Train data is fitted to a suitable classifier upon feature extraction, then once the classifier is trained enough then we predict the results of the test data using the classifier, then compare the original value to the value returned by the classifier.

**Result Analysis:**

Here the accuracy of different classifiers is shown among which the best classifier with the highest accuracy percent is chosen. Some factors such as f score, mean, variance, etc., also account for the consideration of the classifiers.

##### 3.2.1 Use Case Diagram

Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from an external point of view.

**Actors** are external entities that interact with the system. Examples of actors include users like administrators, bank customers…etc., or another system like a central database.

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntacticsemantic and pragmatic rules.

A user-view model system is represented using five different views that describe the system from a distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

This view represents the system from the user’s perspective.

The analysis representation describes a usage scenario from the end-users perspective.

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

USE CASE DIAGRAM

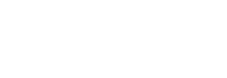
Fig:

-

3.1



Output



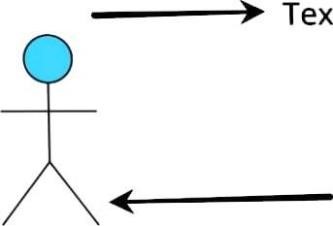
System



Streamlit Interface



Python Backend



###### **Data Flow Diagrams (DFD)**

A data flow diagram is graphical tool used to describe and analyse the movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams.

The physical data flow diagrams show the actual implementation and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. The development of

DFD’S is done in several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called a context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into another process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

##### DFD SYMBOLS

In the DFD, there are four symbols

A square defines a source (originator) or destination of system data.

An arrow identifies data flow. It is the pipeline through which the information flows

An open rectangle is a data store, data at rest or a temporary repository of data

Process that transforms data flow.



|  |
| --- |
|  |

Source or Destination of data

Data flow

##### 0 LEVEL DFD

It is also known as a context diagram. It’s designed to be an abstract view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes.

##### 0 LEVEL DFD

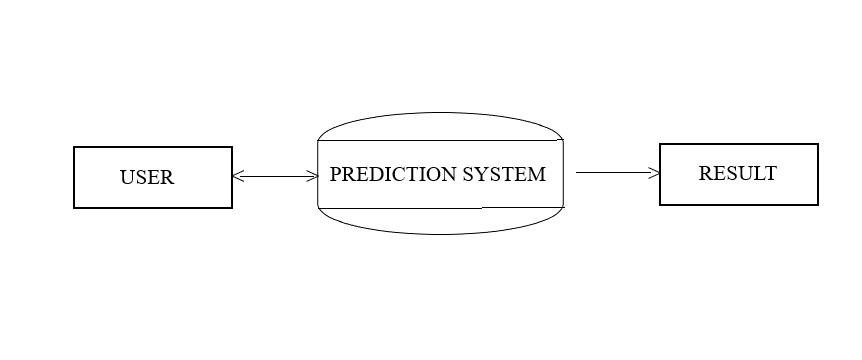


Fig:- 3.2

##### 1 LEVEL DFD

This level provides a more detailed view of the system by breaking down the major processes identified in the level 0 DFD into sub-processes. Each subprocess is depicted as a separate process on the level 1 DFD. The data flows and data stores associated with each sub-process are also shown.

##### 1 LEVEL DFD

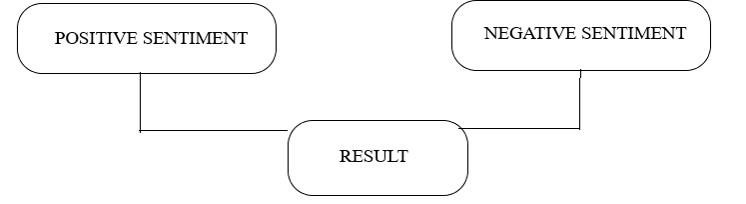
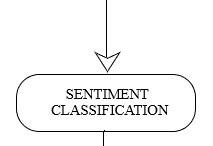
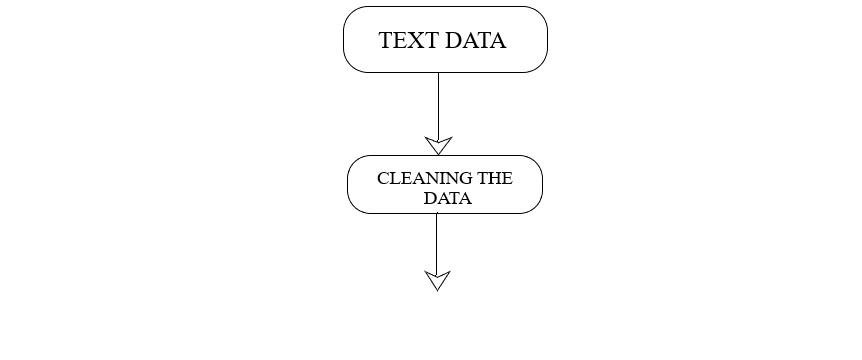


Fig:-3.3

**Chapter 4**

### SOFTWARE CODING, TESTING & IMPLEMENTATION

#### 4.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate text of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a wellplanned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

##### 4.1.1 SYSTEM CODING ENVIRONMENT AND STANDARDS FO**L**LOWED

Different modules specified in the design document are coded in the Coding phase according to the module specification. The main goal of the coding phase is to code from the design document prepared after the design phase through a high-level language and then to unit test this code.

Good software development organizations want their programmers to maintain to some well-defined and standard style of coding called coding standards. They usually make their own coding standards and guidelines depending on what suits their organization best and based on the types of software they develop. It is very important for the programmers to maintain the coding standards otherwise the code will be rejected during code text.

**Purpose of Having Coding Standards:**

A coding standard gives a uniform appearance to the codes written by different engineers.

It improves readability, and maintainability of the code and it reduces complexity also.

It helps in code reuse and helps to detect error easily.

It promotes sound programming practices and increases efficiency of the programmers.

Limited use of globals:

These rules tell about which types of data that can be declared global and the data that can’t be.

Standard headers for different modules:

For better understanding and maintenance of the code, the header of different modules should follow some standard format and information. The header format must contain below things that is being used in various companies:

Name of the module

Date of module creation

Author of the module

Modification history

Synopsis of the module about what the module does

Different functions supported in the module along with their input output parameters

Global variables accessed or modified by the module

**Naming conventions for local variables, global variables, constants and functions:** Some of the naming conventions are given below:

Meaningful and understandable variables name helps anyone to understand the reason of using it.

Local variables should be named using camel case lettering starting with small letter

(e.g., **local Data**) whereas Global variables names should start with a capital letter (e.g., **Global Data**). Constant names should be formed using capital letters only (e.g., **CONSDATA**).

It is better to avoid the use of digits in variable names.

The names of the function should be written in camel case starting with small letters.

The name of the function must describe the reason of using the function clearly and briefly.

Indentation:

Proper indentation is very important to increase the readability of the code. For making the code readable, programmers should use White spaces properly.

Some of the spacing conventions are given below:

There must be a space after giving a comma between two function arguments.

Each nested block should be properly indented and spaced.

Proper Indentation should be there at the beginning and at the end of each block in the program.

All braces should start from a new line and the code following the end of braces also start from a new line.

1. **Error return values and exception handling conventions:**

Functions that encountering an error condition should either return a 0 or 1 for simplify the debugging.

Coding guidelines give some general suggestions regarding the coding style that to be followed for the betterment of understand ability and readability of the code. Some of the coding guidelines are given below:

1. **Avoid using a coding style that is too difficult to understand:**

Code should be easily understandable. The complex code makes maintenance and debugg difficult and expensive.

1. **Avoid using an identifier for multiple purposes:**

Each variable should be given a descriptive and meaningful name indicating the reason behind using it. This is not possible if an identifier is used for multiple purposes and thus it can lead to confusion to the reader. Moreover, it leads to more difficulty during future enhancements.

8**.Code should be well documented:**

The code should be properly commented for understanding easily. Comments regarding the statements increase the understand ability of the code.

1. **Length of functions should not be very large:**

Lengthy functions are very difficult to understand. That’s why functions should be small enough to carry out small work and lengthy functions should be broken into small ones for completing small tasks.

1. **Try not to use GOTO statement:**

GOTO statement makes the program unstructured, thus it reduces the understand ability of the program and also debugging becomes difficult.

**4.1.2 Sample Code Layout**  import subprocess import streamlit as st st.set\_page\_config(page\_title="AboutDevelopers", page\_icon="sent\_icon.png") def launch\_hp():

subprocess.Popen(["streamlit", "run", "Sentomization.py"])

rplink =

"https://drive.google.com/file/d/1HNndFaminUNH19o9bCiFL051lbtsQzZT/vi ew?usp=sharing" rp = "Click to read Research Paper on Data Analysis"

with st.sidebar:

st.image("sent\_logo.png") st.header("Sentomization-Text Data Analysis") st.write("Sentomization is an Online Intelligent Analytical System designed keeping in mind handy and quick text data based Analysis.It has the ability to give quick output of Text Data either by user's Text Input or Extracting text from a PDF File") a=st.button("Return to Home Page") st.markdown(f"[{rp}]({rplink})")

if a:

launch\_hp() col1, col2 = st.columns(2) with col1:

st.title("Vasu Gupta") st.header("Bachelors of Computer Applications Undergraduate") st.subheader("Data Scientist") st.subheader("Business Analyst") st.write("Contact Me:") st.write("vasugupta0112@gmail.com") link\_linkedin = "https://www.linkedin.com/in/vasu-gupta-6aba81218/" link\_text = "LinkedIn" st.markdown(f"[{link\_text}]({link\_linkedin})")

with col2:

st.title("Harshita") st.header("Bachelors of Computer Applications Undergraduate") st.subheader("Cloud Computing Trainee") st.subheader("ML OPS Trainee") st.write("Contact Me:") st.write("harshitac5535@gmail.com") link\_url = "http://www.linkedin.com/in/harshita-759469219" link\_text = "LinkedIn" st.markdown(f"[{link\_text}]({link\_url})")

import streamlit as st import subprocess st.set\_page\_config(page\_title="Sentomization", page\_icon="sent\_icon.png")

with st.sidebar:

st.image("sent\_logo.png") st.header("Text Data Analysis") st.write("These tools are designed to make it easier to analyze, process, and visualize text data, allowing users to extract insights and make data-driven decisions. One of the major advantages of online text data analysis tools is that they are usually easy to use and require little or no programming knowledge. This makes them accessible to a wide range of users, including business analysts, marketers, and researchers who may not have a background in data science. Another advantage of these tools is that they are often cloud-based, which means that they can be accessed from anywhere with an internet connection. This makes it easy to collaborate with others and to share insights and results with stakeholders. Overall, online text data analysis tools are a powerful way to unlock insights from textual data and to make data-driven decisions. As the amount of textual data continues to grow, we can expect these tools to become even more sophisticated and powerful in the years to come.")

col1, col2 = st.columns(2)

with col1:

st.image("wc.jpg",width=400) b1= st.button("Create Wordcloud") with col2:

st.image("sa.jpg",width=400) b2=st.button("Sentiment Analysis")

# Insert a vertical gap between columns st.write("")

col3, col4 = st.columns(2)

with col3:

st.image("barc.jpg",width=400) b3=st.button("Bar Chart")

with col4:

st.image("aboutus.jpg",width=400,) b4=st.button("About")

def launch\_wc():

subprocess.Popen(["streamlit", "run", "WordCloudGenerator.py"])

def launch\_sa():

subprocess.Popen(["streamlit", "run", "sentimentAnalyzer.py"]) def launch\_barc():

subprocess.Popen(["streamlit", "run", "BarChartGenerator.py"]) def launch\_about():

subprocess.Popen(["streamlit","run","About.py"])

# Create button to launch App 2 if b1:

launch\_wc() if b2:

launch\_sa() if b3:

launch\_barc() if b4:

launch\_about() import subprocess import streamlit as st

import PyPDF2 from textblob import TextBlob st.set\_page\_config(page\_title="SentimentAnalyzer", page\_icon="sent\_icon.png") with st.sidebar:

st.image("sent\_logo.png") st.header("Sentiment Analysis") st.write("Positive sentiment analysis examines text data to identify and understand expressions of joy, satisfaction, and optimism, helping businesses gauge customer happiness and enhance their offerings. Negative sentiment analysis analyzes textual content to detect instances of frustration, dissatisfaction, or disappointment, enabling organizations to address customer concerns, improve products, and enhance overall customer experience.") st.title("Sentiment Analyzer") st.subheader("Upload a PDF file or give Text input for Sentiment Analysis") input=st.text\_area("Enter Text Input:",value="") uploaded\_file = st.file\_uploader("Choose a PDF file", type="pdf") b1=st.button("Click to generate Sentiment of Text") hp=st.button("Home Page") def launch\_hp():

subprocess.Popen(["streamlit", "run", "Sentomization.py"])

if hp:

launch\_hp() def text\_extract(data):

pdf\_reader = PyPDF2.PdfFileReader(data)

pdf\_text = "" for i in range(pdf\_reader.getNumPages()):

pdf\_text += pdf\_reader.getPage(i).extractText() return pdf\_text def prediction(text): blob = TextBlob(text) sentiment = blob.sentiment.polarity

# Output sentiment result if sentiment > 0: a="Sentiment: Positive" elif sentiment < 0:

a="Sentiment: Negative"

else:

a="Sentiment: Neutral" return a if b1: if uploaded\_file is not None: inp=text\_extract(uploaded\_file) outp=prediction(inp) st.subheader(outp)

elif input is not None: outp2=prediction(input) st.subheader(outp2)

import streamlit as st from wordcloud import WordCloud import matplotlib.pyplot as plt import subprocess import PyPDF2 def launch\_hp():

subprocess.Popen(["streamlit", "run", "Sentomization.py"]) def create\_wordcloud(topic):

wordcloud = WordCloud().generate(topic) fig, ax = plt.subplots(figsize = (16, 9)) ax.imshow(wordcloud)

plt.axis("off") return st.pyplot(fig)

def text\_extract(data):

pdf\_reader = PyPDF2.PdfFileReader(data) pdf\_text = "" for i in range(pdf\_reader.getNumPages()):

pdf\_text += pdf\_reader.getPage(i).extractText() return pdf\_text

st.set\_page\_config(page\_title="WordCloudGenerator", page\_icon="sent\_icon.png") with st.sidebar:

st.image("sent\_logo.png") st.header("WordCloud") st.write("A word cloud visually represents the frequency of words in a text, with more frequently occurring words appearing larger, making it a useful tool for quickly identifying key themes or topics in a document or dataset.")

# Add title and description st.title("Word Cloud Generator") st.subheader("Upload a PDF file or give Text input for Word Cloud Creation") input=st.text\_area("Enter Text Input:",value="") uploaded\_file = st.file\_uploader("Choose a PDF file", type="pdf") b1=st.button("Click to generate Word Cloud") hp=st.button("Home Page")

if b1: if uploaded\_file is not None: a=text\_extract(uploaded\_file)

# Open the PDF file and extract its text content

# Display the extracted text create\_wordcloud(a)

elif input is not None:

create\_wordcloud(input) if hp:

launch\_hp()

import streamlit as st from collections import Counter import matplotlib.pyplot as plt import PyPDF2 import subprocess

def launch\_hp():

subprocess.Popen(["streamlit", "run", "Sentomization.py"])

# Get input text from user def barc(input\_text):

# Convert input text to lowercase and split into words words = input\_text.lower().split()

# Calculate word frequencies word\_freqs = Counter(words)

# Get 10 most common words top\_words = word\_freqs.most\_common(20)

# Create bar chart of word frequencies fig, ax = plt.subplots(figsize=(16,9)) ax.bar([w[0] for w in top\_words], [w[1] for w in top\_words]) ax.set\_xlabel("Word") ax.set\_ylabel("Frequency") ax.set\_title("Top 10 Words by Frequency")

# Display bar chart in Streamlit app return st.pyplot(fig)

def text\_extract(data):

pdf\_reader = PyPDF2.PdfFileReader(data) pdf\_text = "" for i in range(pdf\_reader.getNumPages()):

pdf\_text += pdf\_reader.getPage(i).extractText() return pdf\_text

# Set page layout

st.set\_page\_config(page\_title="Bar Chart Generator", page\_icon="sent\_icon.png") with st.sidebar:

st.image("sent\_logo.png") st.header("Bar Chart") st.write("Bar Chart in Text Data Analysis is used to check the Most Frequently occuring words in a Dataset.") # Add title and description st.title("Bar Chart Generator")

st.subheader("Upload a PDF file or give Text input for Bar Chart Creation") input=st.text\_area("Enter Text Input:",value="") uploaded\_file = st.file\_uploader("Choose a PDF file", type="pdf") b1=st.button("Click to generate Bar Chart") hp=st.button("Home Page")

if b1: if uploaded\_file is not None: a=text\_extract(uploaded\_file)

# Open the PDF file and extract its text content

# Display the extracted text barc(a)

elif input is not None:

barc(input)

if hp:

launch\_hp()

#### 4.2 TESTING

Well, it should be error-free. If testing is done successfully it will remove all the errors from the software.

Principles of Testing:-

1. All the tests should meet the customer requirements.
2. To make our software testing should be performed by a third party.
3. Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.

(iv)All the tests to be conducted should be planned before implementing it.

(v) It follows the 80/20 rule which states that 80% of errors come from 20% of program Start testing with small parts and extend it to large parts.

##### 4.2.1 OVERVIEW & APPROACH

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Target of the test are -

**Errors** - These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.

**Fault** - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.

**Failure** - failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

Tests can be conducted based on two approaches –

Functionality testing

Implementation testing

When functionality is being tested without taking the actual implementation in concern it is known as black-box testing. The other side is known as white-box testing where not only functionality is tested but the way it is implemented is also analysed.

Exhaustive tests are the best-desired method for perfect testing. Every single possible value in the range of the input and output values is tested. It is not possible to test each and every value in real-world scenario if the range of values is large.

**Black-box testing**

It is carried out to test functionality of the program. It is also called ‘Behavioral’ testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested ‘ok’, and problematic otherwise.

In this testing method, the design and structure of the code are not known to the tester, and testing engineers and end users conduct this test on the software.

**White-box testing**

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as ‘Structural’ testing.

In this testing method, the design and structure of the code are known to the tester.

Programmers of the code conduct this test on the code.

The below are some White-box testing techniques:

**Control-flow testing** - The purpose of control-flow testing to set up test cases which cover all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.

**Data-flow testing** - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

#### 4.3 TEST PLAN

A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverables.

The test plan is a base of every software's testing. It is the most crucial activity which ensures availability of all the lists of planned activities in an appropriate sequence.

The test plan is a template for conducting software testing activities as a defined process that is fully monitored and controlled by the testing manager.

##### 4.3.1 FEATURES TO BE TESTED & NOT TO BE TESTED

Features to be tested:

This part covers what the majority imagines a Test Plan should present – a more precise list of features to inspect during a specified time. For example, we mentioned the checkout functionality in Test Items. In Features to Be Tested, we should list separate components of the flow: entering delivery information, choosing a payment method, order confirmation, etc. As for the timing, a client and a QA company discuss it before documentation writing.

Features not to be tested:

Here, you list the features that a QA team is not going to test for a particular reason. It doesn’t matter why you don’t cover this scope. Just don’t forget to state what features remain out of your tasks and are a client’s responsibility.

#### 4.4. TEST CASES

Testing itself may be defined at various levels of SDLC. The testing process runs parallel to software development. Before jumping to the next stage, a stage is tested, validated, and verified.

Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels -

**Unit Testing**

While coding, the programmer performs some tests on that unit of the program to know if it is error-free. Testing is performed under the white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

**Integration Testing**

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passing and data updating, etc.

**System Testing**

The software is compiled as text and then it is tested as a whole. This can be accomplished using one or more of the following tests:

**Functionality testing** - Tests all functionalities of the software against the requirement.

**Performance testing** - This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.

**Security & Portability** - These tests are done when the software is meant to work on various platforms and accessed by number of persons.

**Acceptance Testing**

When the software is ready to hand over to the customer it has to go through the last phase of testing where it is tested for user interaction and response. This is important because even if the software matches all user requirements and if the user does not like the way it appears or works, it may be rejected.

**Alpha testing** - The team of developers themselves perform alpha testing by using the system as if it is being used in the work environment. They try to find out how a user would react to some action in software and how the system should respond to inputs.

**Beta testing** - After the software is tested internally, it is handed over to the users to use it under their texture environment only for testing purposes. This is not as yet the delivered text. Developers expect that users at this stage will bring minute problems, which were skipped to attend.

**Regression Testing**

Whenever a software text is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code.

This is known as regression testing.

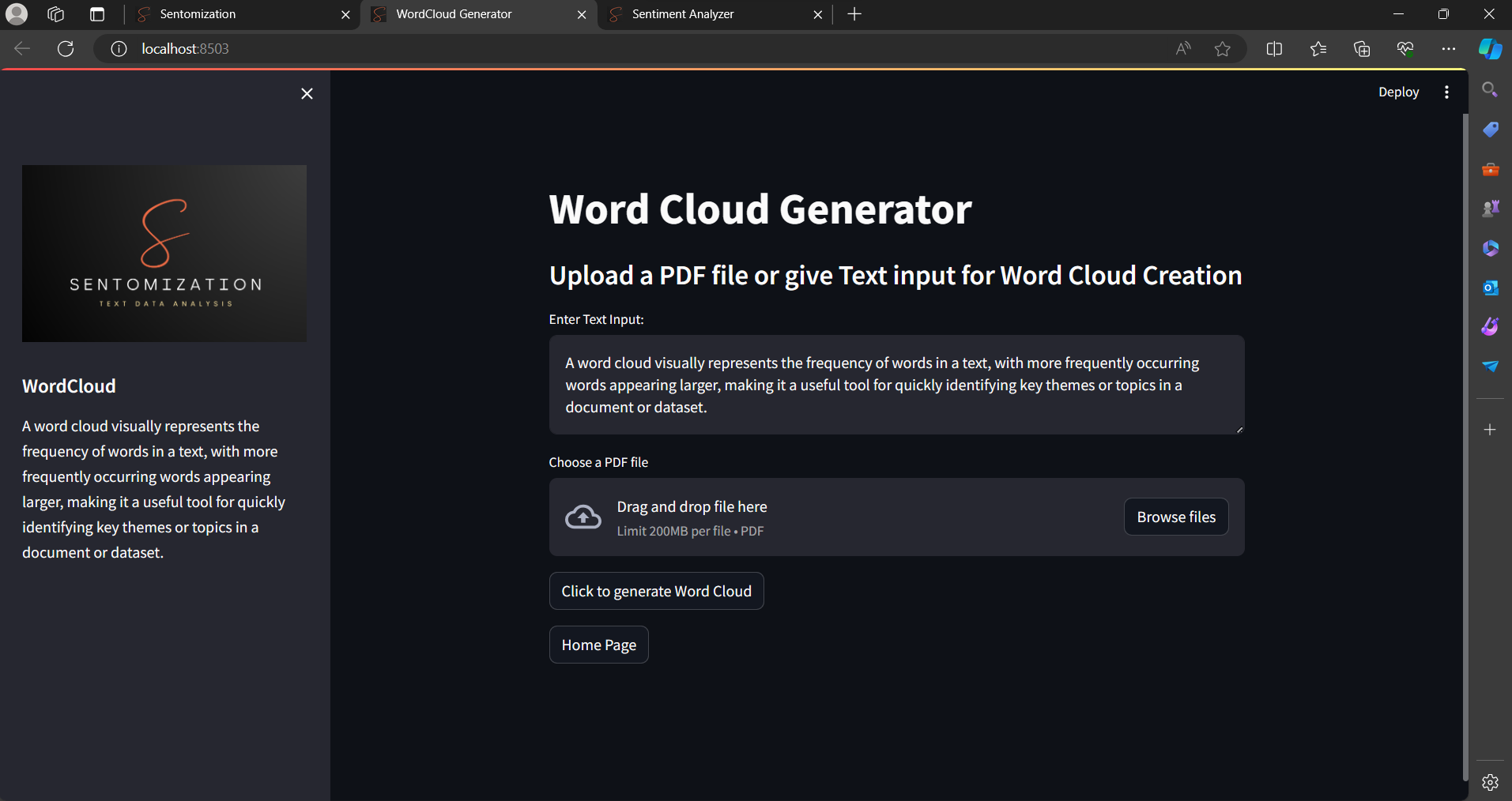
**CHAPTER 5**

### OUTPUT FORMS & REPORTS

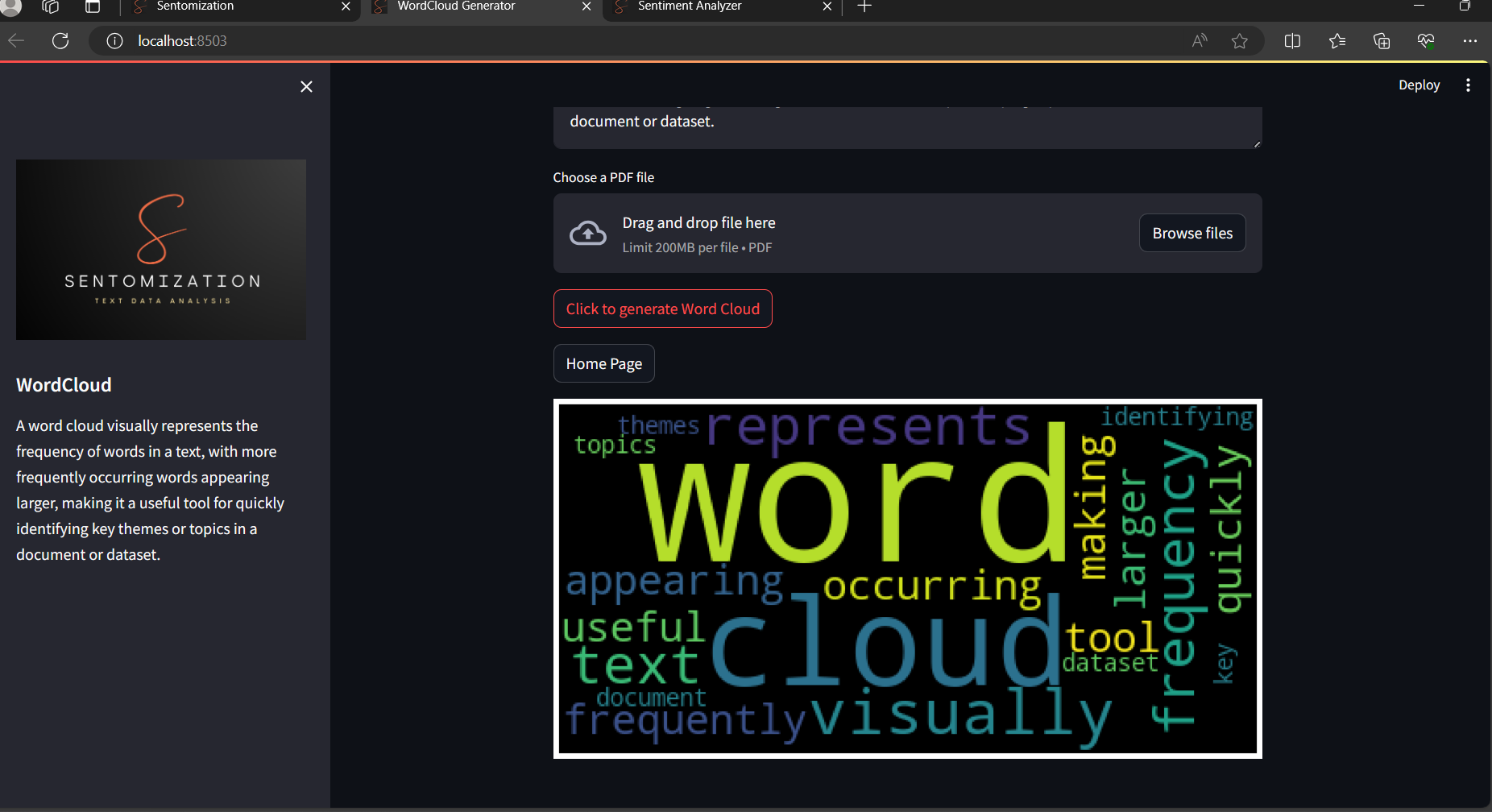
**5.1 word cloud**

User interface

Input

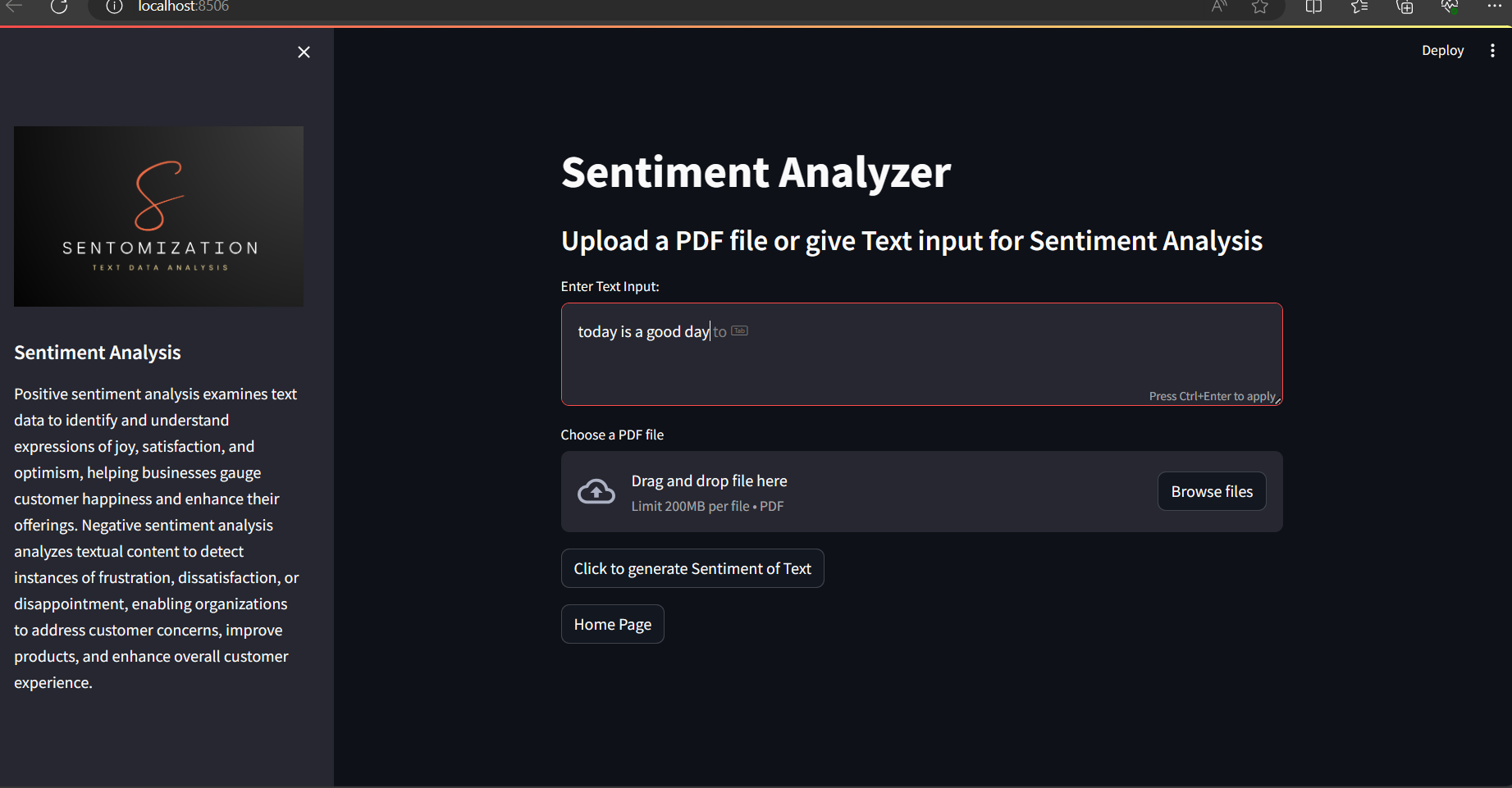


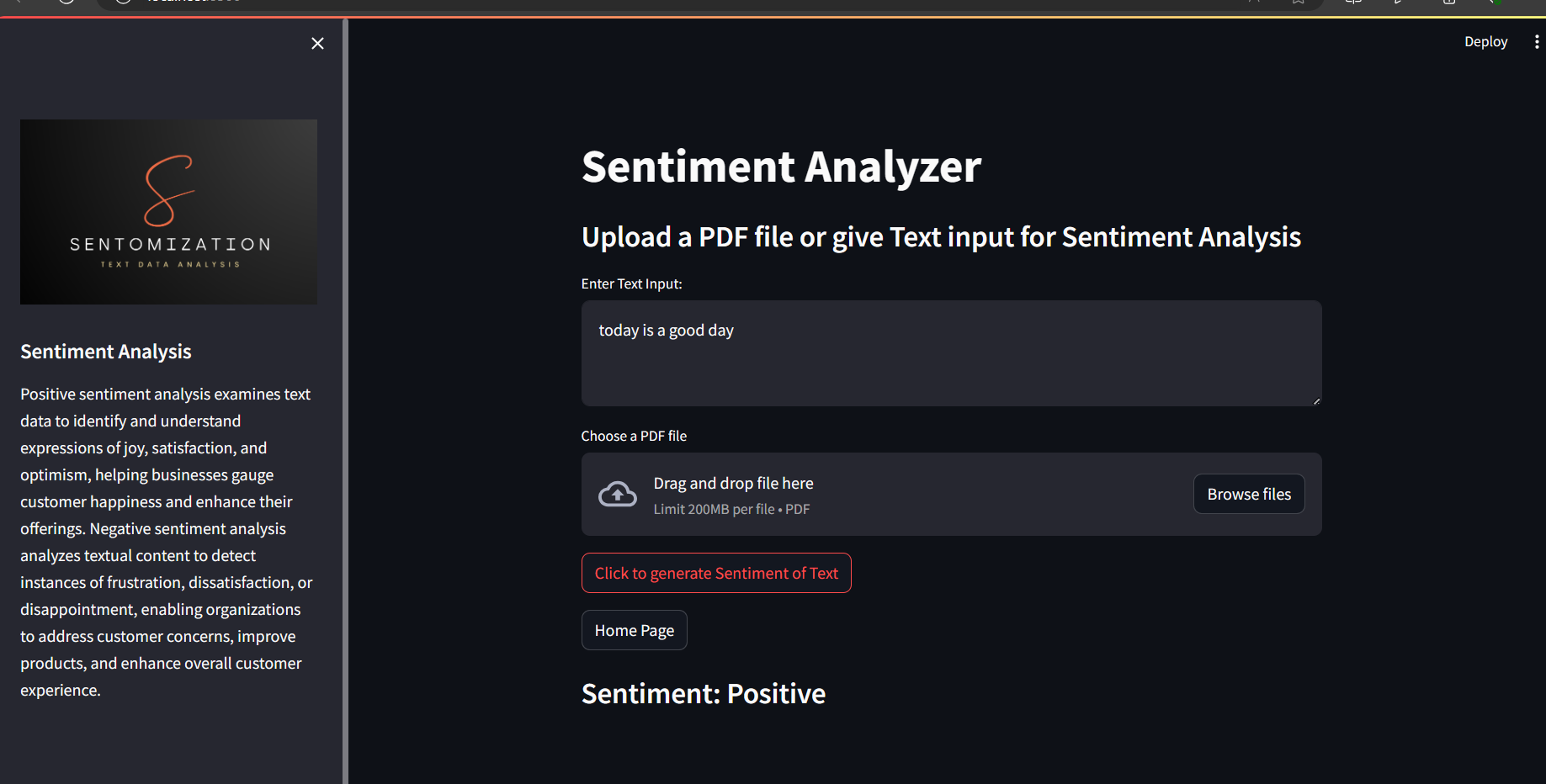
Output

****

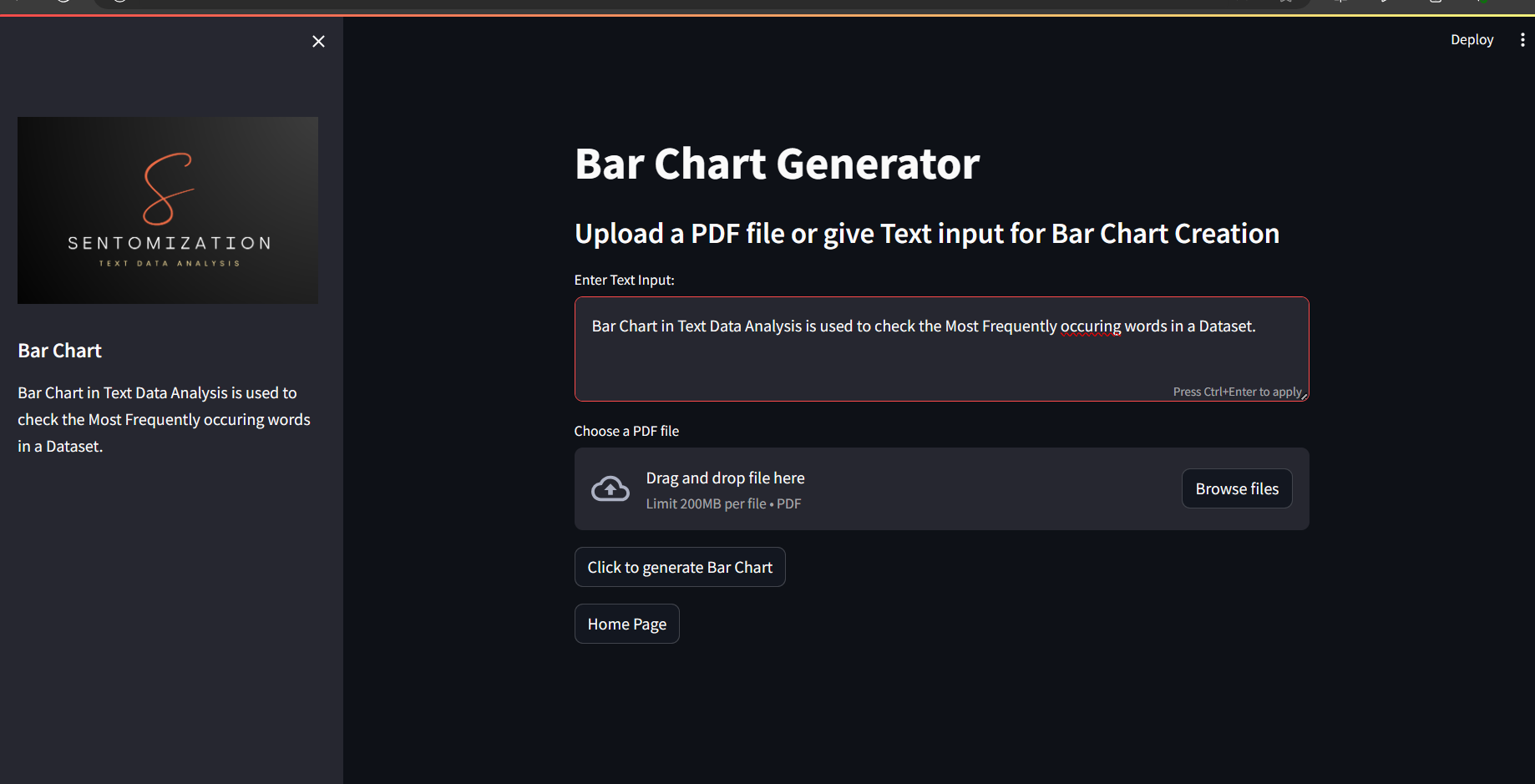
**Sentiment analyser**

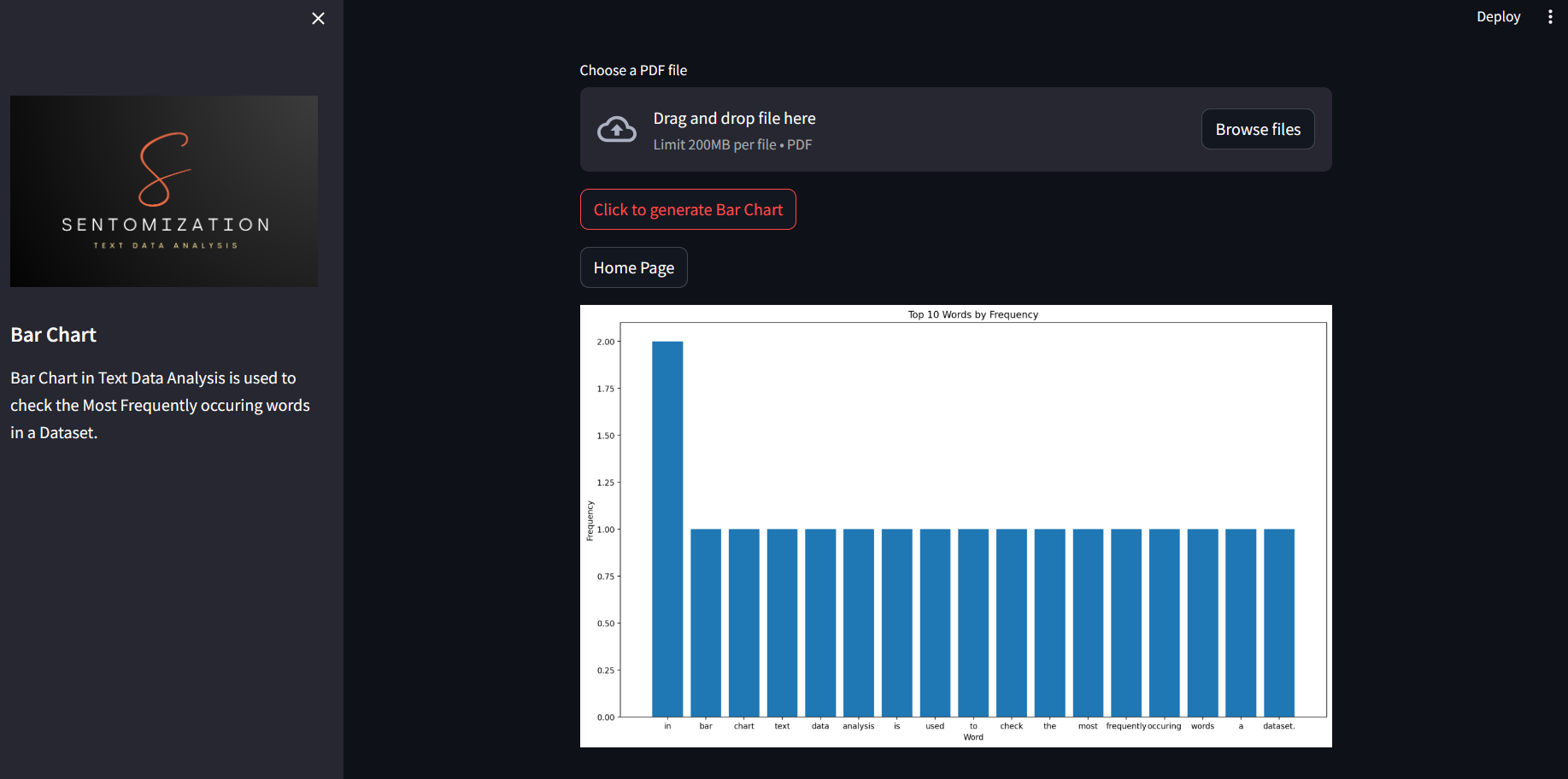
**input**

****

****

**Bar Chart Generator**

****

****

**CHAPTER 6**

**CONCLUSION & FUTURE ENHANCEMENT(S)**

##### 6.1 CONCLUSION

Text data analysis is a powerful tool for businesses to understand customer text and provide better customer experiences. It can be used to identify customer feedback, categorize customer texts, and measure the success of marketing campaigns. By using text data analysis, companies can gain valuable insights into their customers’ feelings and opinions about their texts or services. This can help them make more informed decisions and develop better strategies for engaging with their customers. With the right approach and tools, text data analysis can be an invaluable asset in helping companies understand their customers better and build stronger relationships with them.

It can be used to gain insights into customer opinions and preferences, which can then be used to improve customer experience and satisfaction. Text data analysis can also help businesses identify potential areas of improvement and opportunities for growth. By analyzing customer sentiment, businesses can better understand their target audience and make well-informed decisions. In conclusion, text data analysis is an invaluable tool that should be utilized by businesses in order to gain a better understanding of their customers.

* The skills that are essential for developing this project.
* The planning that goes into implementing a project.
* The importance of proper planning and an organized methodology.
* The key element of team spirit and co-ordination in a successful project.

The project also helped us in gaining practical knowledge of not only programming with web-based application development. This will provide better opportunities and guidance in future for developing projects independently.

##### 6.2 FUTURE SCOPE

Text data analysis has become an important tool for businesses in order to understand how their customers feel about their texts and services. By analyzing the text of customer feedback, businesses can gain valuable insights that can help them make better decisions and improve customer experience.

The future scope of text data analysis is very promising. With the growth of social media, businesses have access to a vast amount of data that they can use to analyze customer sentiment. This data can be used to identify trends in customer behavior and develop strategies that will meet their needs. Additionally, AI-powered text data analysis tools are becoming increasingly sophisticated and accurate, allowing businesses to get more accurate insights into customer text than ever before.

Text data analysis is a powerful tool that allows businesses to understand the feelings and emotions of their customers. It can be used to measure customer satisfaction, evaluate customer feedback, and discover insights into customer behavior. With the help of text data analysis, companies can gain valuable information about their customers’ experiences with their texts or services and make better decisions in order to improve them. The future scope of text data analysis looks very promising as more and more companies are turning towards it to gain insights into their customers' needs and preferences. Text data analysis is being used in various industries such as retail, healthcare, finance, travel & hospitality, and more. It promises to revolutionize the way companies interact with their customers by providing them with real-time data about customer sentiment.

#### **BIBLIOGRAPHY**

FOR INFORMATION GATHERING:

<https://www.nltk.org/><https://www.w3schools.com/python/pandas/default.asp><https://en.m.wikipedia.org/wiki/Sentiment_analysis>[https://sciki](https://scikit/)[t-learn.org/stable/](https://scikit-learn.org/stable/)