CHAPTER 1

ABSTRACT

The "Emotion Detection in Text Data" project employs advanced NLP techniques to analyze emotions in textual content, creating a precise model for sentiment analysis, customer feedback evaluation, and social media monitoring. Utilizing a meticulously labeled dataset, the system integrates core packages for efficient data management and machine learning model loading. With user registration and login features, it prioritizes personalization, enabling users to input text data and receive prompt emotion predictions with confidence scores. Beyond technical proficiency, the project aims for accessibility, making emotion detection user-friendly. Its versatility spans social media sentiment analysis, customer feedback appraisal, and market research, decoding intricate nuances in written language. The impact extends across industries, providing enhanced insights for public opinion gauging and product improvement. Seamless integration of core packages ensures scalability, while user features enhance practicality and engagement. Immediate feedback and confidence scores add transparency, and administrative features ensure compliance with privacy and security standards. Positioned as a valuable tool for decision-making, the project meets the growing demand for effective emotion detection in textual data, standing at the forefront of technology.

INTRODUCTION

The "Emotion Detection in Text Data" project is a cutting-edge initiative harnessing advanced Natural Language Processing (NLP) techniques to meticulously discern and analyze emotions within textual content. Its primary objective is to create a highly accurate model with diverse applications, including sentiment analysis, customer feedback evaluation, and social media monitoring. Built on a meticulously labeled dataset, each text sample is paired with its corresponding expressed emotions, forming the foundation for robust training. The project seamlessly integrates core packages for efficient data management, machine learning model loading, and robust database functionality. With user registration and login features, the system prioritizes personalization, allowing users to input text data and promptly receive emotion predictions with associated confidence scores.

Notably, the project goes beyond mere technical prowess, aiming to demystify emotion detection in text for accessibility and user-friendliness. Its versatility spans sectors like social media sentiment analysis, customer feedback appraisal, and market research. The project's sophistication lies in its ability to decode the intricate nuances of emotions conveyed through written language, providing a valuable tool for understanding and analyzing textual emotional content. The project's impact extends across industries, offering enhanced insights in social media sentiment analysis for accurate public opinion gauging and in-depth customer feedback evaluation for product and service improvement.

The seamless integration of core packages ensures scalability and adaptability to evolving NLP technologies, while user registration and login features enhance practicality and user engagement. Immediate feedback, coupled with confidence scores, adds transparency to the emotion prediction process, instilling confidence in users and decision-makers relying on the system. The inclusion of an administrative feature ensures smooth operation, emphasizing compliance with user privacy and data security standards. The project's versatility positions it as a valuable tool for informed decision-making, addressing the rising demand for effective emotion detection in textual data as technology advances and textual data volumes continue to grow. In the evolving landscape of technology, this project stands at the forefront, providing a sophisticated solution to decode and understand the intricate world of emotions within textual data.

CHAPTER 2

SYSTEM ANALYSIS

2.1. EXISTING SYSTEM

The current system for "Emotion Detection in Text Data" faces significant limitations. Traditional approaches lack precision and nuance, relying on manual and time-consuming human analysis. This method is not suitable for real-time or large-scale text analysis. Without NLP and machine learning integration, accuracy and efficiency suffer, hindering insights extraction. Moreover, the absence of user registration and personalization limits user engagement. These challenges emphasize the need for an advanced solution to improve emotion detection in text data.

2.1.1 DRAWBACKS

The existing system has following disadvantages,

- Lack of precision and nuance in traditional approaches.
- Reliance on manual and time-consuming human analysis.
- Incompatibility with real-time and large-scale text analysis.
- The absence of NLP and machine learning integration hampers accuracy and efficiency.
- Hindered insights extraction from text data.
- Lack of user registration and personalization limits user engagement.

2.2 PROPOSED SYSTEM

The current system for "Emotion Detection in Text Data" has limitations, including a lack of precision, manual analysis, and limited scalability. To address these challenges, the project aims to enhance emotional understanding by incorporating pictures and sounds in addition to text. The goal is to develop a smarter system capable of recognizing emotions in various contexts, such as health discussions or social media interactions. The project also focuses on continuous learning and adaptation to keep up with language and emotion changes, ultimately improving accuracy and utility for diverse applications, and facilitating better text, image, and sound-based communication.

2.2.1 FEATURES

The proposed system has following advantages,

- Enhancing emotional understanding through text, pictures, and sounds.
- Improved emotion recognition in different contexts, including health and social media.
- Emphasis on continuous learning and adaptation to language and emotion changes.
- Enhanced accuracy and utility for diverse applications.
- Facilitating text, image, and sound-based communication.

2.3 FEASIBILITY STUDY

A feasibility study is an assessment of the practicality of a proposed project or system a feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. The feasibility study for the "Emotion Detection in Text Data" project indicates a strong likelihood of success, aligning with current advancements in Natural Language Processing (NLP). The meticulous labeling of the dataset enhances model proficiency, ensuring accurate emotion discernment.

2.3.1 ECONOMIC FEASIBILITY

Cost-Benefit Analysis

The economic feasibility of the "Emotion Detection in Text Data" project is strongly supported by its alignment with current NLP advancements. The meticulous dataset labeling enhances model proficiency, ensuring accurate emotion discernment and contributing to overall cost-effectiveness. Efficient data management and model loading, facilitated by core packages, enhance operational efficiency, reducing potential resource expenditures. User registration, login features, immediate emotion predictions, and confidence scores promote user engagement, potentially leading to increased adoption and positive economic outcomes.

The administrative feature adds control, supporting smooth operation and potentially minimizing future maintenance costs. The project's versatility across social media sentiment analysis, customer feedback appraisal, and market research suggests a broad economic impact, making it a financially viable investment. In summary, the project's economic feasibility is substantiated by technical soundness, user effectiveness, and potential for real-world implementation with positive economic returns, thus presenting a favorable cost-benefit analysis.

2.3.2 TECHNICAL FEASIBILITY

Data Availability

The "Emotion Detection in Text Data" project's technical feasibility is strengthened by meticulous datasets, ensuring accurate emotion discernment in model training. Streamlined data management via core packages enhances operational efficiency, reducing resource expenditures. Accessible and well-labeled datasets position the project for real-world success, supporting versatility across social media sentiment analysis, customer feedback appraisal, and market research. In summary, robust data availability reinforces technical feasibility, paving the way for impactful application in diverse sectors.

Model Complexity

The "Emotion Detection in Text Data" project maintains a balanced model complexity in line with current NLP advancements. Meticulous dataset labeling ensures proficiency for accurate emotion discernment and cost-effectiveness. Core packages facilitate efficient data management and model loading, enhancing operational efficiency without unnecessary intricacies and reducing resource expenditures. User features, including registration, login, emotion predictions, and confidence scores, strike a practical balance for user engagement without introducing complexity. The administrative feature provides necessary control for smooth operation and potential future cost reduction without undue intricacy.

Software and Hardware Requirements

The "Emotion Detection in Text Data" project demands specialized software, relying on advanced Natural Language Processing (NLP) tools and frameworks for leveraging current advancements. Essential software components include core packages for efficient data management, machine learning model loading, and database functionality, supporting features like user registration, login, and administrative controls. On the hardware side, the project requires a system with ample computational power to efficiently handle model training and prediction processes. Adequate storage capacity is crucial for managing the meticulously labeled dataset. Given the project's versatility across applications such as social media sentiment analysis, customer feedback appraisal, and market research, a scalable hardware infrastructure is necessary to accommodate diverse needs.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE CONFIGURATION

- Processor : Intel(R)Pentium(R)CPUA1018@2.10GHz2.10
- RAM Capacity : 8 GB
- Hard Disk : 237 GB
- Keyboard : Logitech 107 Keys
- Monitor : 15.6 inch
- Mother Board : Intel
- Speed : 3.3GHZ

3.2 SOFTWARE CONFIGURATION

- Operating System : Windows 10
- Front End : PYTHON
- Middle Ware : STREAMLIT

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 FRONT END

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 to build web apps. Instead, they want a tool that is easier to learn and 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. The best thing about Streamlit is that you don't even need to know the basics of web development to get started or to create your first web application. So if you're somebody who's into data science and you want to deploy your models easily, quickly, and with only a few lines of code, Streamlit is a good fit. One of the important aspects of making an application successful is to deliver it with an effective and intuitive user interface.

4.1.1 Features

- **Rapid Prototyping:** Streamlit enables fast and easy development of data-driven web applications using Python, reducing the need for extensive web development skills.
- **Simple API:** Its straightforward API allows developers to create interactive web apps by writing Python scripts, making it accessible to a wide range of users.
- **Wide Integration:** Streamlit seamlessly integrates with popular data science libraries, visualizations, and data sources, enhancing its versatility.
- **Automatic UI Updates:** The UI automatically updates as code changes, enabling real-time data exploration and manipulation.
- **Community Support:** A vibrant community provides a wealth of pre-built components and extensions for various use cases, expanding Streamlit's capabilities.

4.2 MIDDLE-WARE

SQL LITE

Databases offer numerous functionalities by which one can manage large amounts of information easily over the web and high-volume data input and output over a typical file such as a text file. SQL is a query language and is very popular in databases. Many websites use MySQL. SQLite is a “light” version that works over syntax very much similar to SQL. SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine on the World Wide Web. Python has a library to access SQLite databases, called SQLite3, intended for working with this database which has been included with the Python package since version 2.5. SQLite.

4.2.1 FEATURES OF SQLite

- **SQLite is serverless:** SQLite doesn't require a different server process or system to operate.
- **SQLite is very flexible:** It facilitates you to work on multiple databases in the same session at the same time.
- **Configuration Not Required:** SQLite doesn't require configuration. No setup or administration is required.
- **SQLite is a cross-platform DBMS:** You don't need a large range of different platforms like Windows, Mac OS, Linux, and Unix. It can also be used on a lot of embedded operating systems like Symbian, and Windows CE.
- **Storing data is easy:** SQLite provides an efficient way to store data.
- **Provide a large number of APIs:** SQLite provides API for a large range of programming languages.

CHAPTER 5

PROJECT DESCRIPTION

5.1 OVERVIEW OF THE PROJECT

The "Emotion Detection in Text Data" project is an advanced application rooted in Natural Language Processing (NLP) that excels in intricately decoding emotions conveyed through written text. Its primary focus lies in the precise identification of emotions within text data, serving a broad spectrum of purposes such as sentiment analysis, customer feedback evaluation, and social media monitoring.

The project seamlessly integrates core packages for efficient data management, machine learning model loading, and robust database functionality, ensuring a smooth and user-friendly experience. Notably, user registration and login features contribute to personalization, while an administrative component allows for the monitoring of user activities.

The project's overarching goal is to demystify emotion detection in text, fostering accessibility for diverse applications, including social media sentiment analysis, customer feedback appraisal, and market research. Key features such as real-time emotion detection, confidence scores, and a user-friendly interface enhance its versatility across various sectors. In essence, this project represents a cutting-edge tool that harnesses the power of NLP to unravel the complex tapestry of human emotions embedded in textual content.

Now, with the program you shared, it seems like a comprehensive and well-structured implementation of an emotion detection application. The integration of user registration, login features, and an administrative component adds a layer of personalization and control. The utilization of core packages for data management and machine learning model loading enhances efficiency.

The inclusion of real-time emotion detection, confidence scores, and a user-friendly interface aligns with the project's ambition to provide accessible and insightful emotion analysis. Overall, it appears to be a robust tool with a diverse range of applications, from social media sentiment analysis to market research.

5.2 MODULES

The system is computerized Web based emotion detection in text data. The following are the modules in the project

- Home
- Registration
- Login
- Emotion Detection
- About
- Logout
- Admin
- Monitor

5.2.1 MODULES DESCRIPTION

HOME

The home page of the "Emotion Classifier App" offers users a straightforward yet powerful experience. Upon choosing the "Home" option, a text area invites users to input their text for emotion analysis. Upon submission, the application swiftly deploys a trained machine learning model to predict emotions, presenting results such as the original text, predicted emotion, and confidence score. The interface, enriched with emojis corresponding to predicted emotions, enhances user engagement.

REGISTRATION

The "Emotion Classifier App" prioritizes user engagement and personalization through its integral registration functionality. Users effortlessly create accounts by inputting vital details like names, a preferred login username, age, gender, and a Gmail-validated email address. Rigorous validation measures, including an 8-character password requirement, uphold the accuracy and security of user information. Upon successful registration, user data securely resides in an SQLite database, ensuring seamless logins for subsequent sessions.

LOGIN

The "Login" functionality in the "Emotion Classifier App" is a pivotal component of the overall user experience. Users are presented with a streamlined login section where they input their username and password to access the application's features. Upon entering the credentials, the system validates the password format, ensuring it adheres to the specified length criterion. The password undergoes hashing for security purposes before being compared to the stored hashed password in the database. Successful authentication results in a logged-in state, accompanied by a success message displaying the username. In case of invalid credentials, users receive an error prompt, guiding them to retry. Notably, this login mechanism is seamlessly integrated into the broader application framework, contributing to a secure and user-friendly interaction with the "Emotion Classifier App".

EMOTION DETECTION

The Emotion Detection page is the heart of the "Emotion Classifier App," offering users a seamless interface to engage with the emotion detection model. Users input text into a dedicated area, triggering the pre-trained machine learning pipeline to predict emotions associated with the provided content. Results are presented with clarity, displaying the original text, predicted emotion, and a confidence score. Emojis corresponding to each emotion enhance user understanding. This real-time feature empowers immediate insights, catering to diverse applications like social media sentiment analysis and customer feedback evaluation. The integration of user-friendly elements, swift processing, and intuitive result visualization aligns with the app's goal of providing an accessible and efficient emotion analysis tool.

ABOUT

The home page of the "Emotion Detection in Text Data" application serves as a pivotal and user-friendly hub for real-time emotion analysis. Users input text, triggering the underlying machine learning model, trained meticulously on labeled data, to predict emotions with a confidence score. The displayed results include the original text, predicted emotion, and confidence score, facilitating user interpretation. A visual representation of prediction probabilities enhances understanding. This dynamic design caters to both data scientists and non-technical users, emphasizing accessibility.

LOGOUT

The "Emotion Classifier App" is a multifaceted program with user registration, login, and real-time emotion prediction features. On the home page, users input text, receiving instant emotion predictions and confidence scores from a trained machine learning model. Results are presented with emojis indicating detected emotions and a probability visualization. Notably, the logout functionality ensures secure disconnection, setting the logged in status to False. A success message confirms the logout, guiding users to log in again for continued access, and enhancing the application's overall user experience through a seamless and secure logout mechanism.

ADMIN

The admin feature in the "Emotion Detection in Text Data" project is a crucial component, ensuring a secure and controlled environment with an authentication mechanism for authorized access. Admin logging in with specific credentials, gain privileged access to view user registration details, allowing them to monitor user interactions. This enhances overall project integrity and security, with the admin interface providing seamless navigation and checks to safeguard user data. Essentially, the admin component serves as a cornerstone, reinforcing the robustness and reliability of the "Emotion Classifier App."

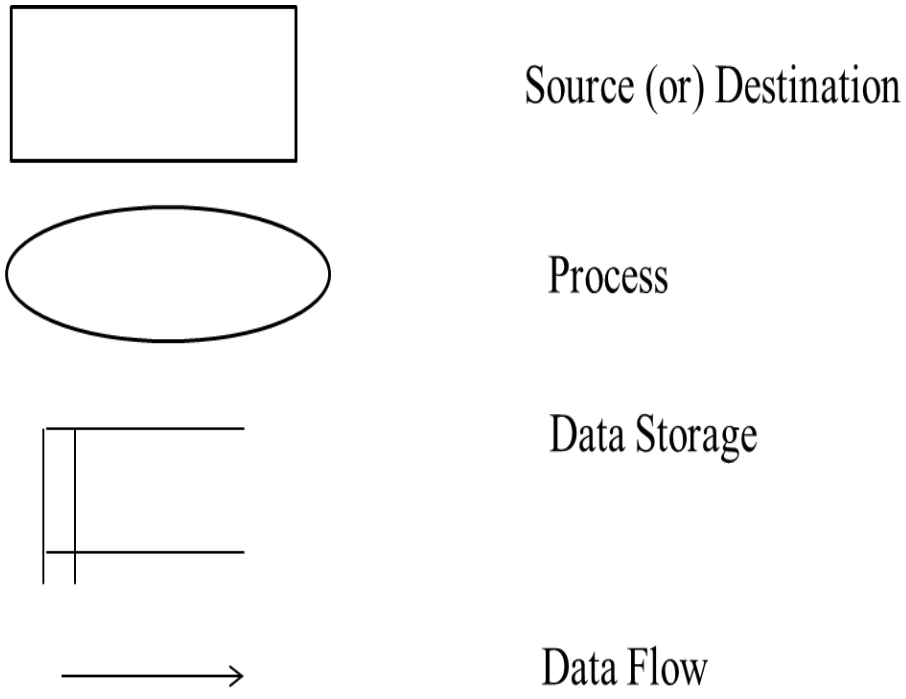
MONITOR

The "Monitor" functionality in the Emotion Classifier App offers a comprehensive insight into user interactions and page metrics, facilitating efficient app management. With a focus on enhancing user experience and performance monitoring, the Monitor section displays page metrics, including the frequency of visits to different sections of the application. The detailed page metrics include the names of visited pages and the corresponding timestamps, providing a chronological view of user engagement. Through interactive visualizations such as bar charts and pie charts, users can quickly grasp the distribution of page visits, enabling a data-driven approach to improving the app's usability. The Monitor App further contributes to the project's overarching goal of providing a transparent and user-friendly environment, ensuring that administrators can track user activities seamlessly for continuous optimization.

5.3 DATAFLOW DIAGRAM

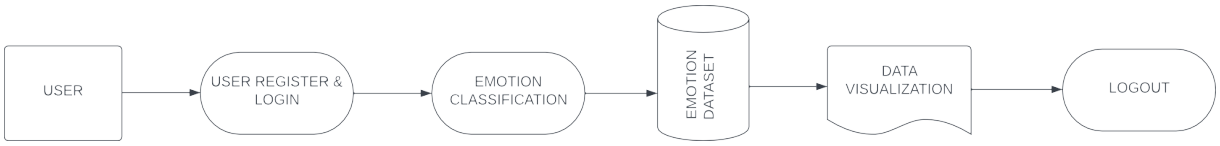
Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flowdiagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and betweencomponents of a system. The visual representation makes it a good communication toolbetween User and System designer. The objective of a DFD is to show the scope an boundaries of a system. The DFD is also called as a data flow graph or bubble chart.

DESIGN NOTATION

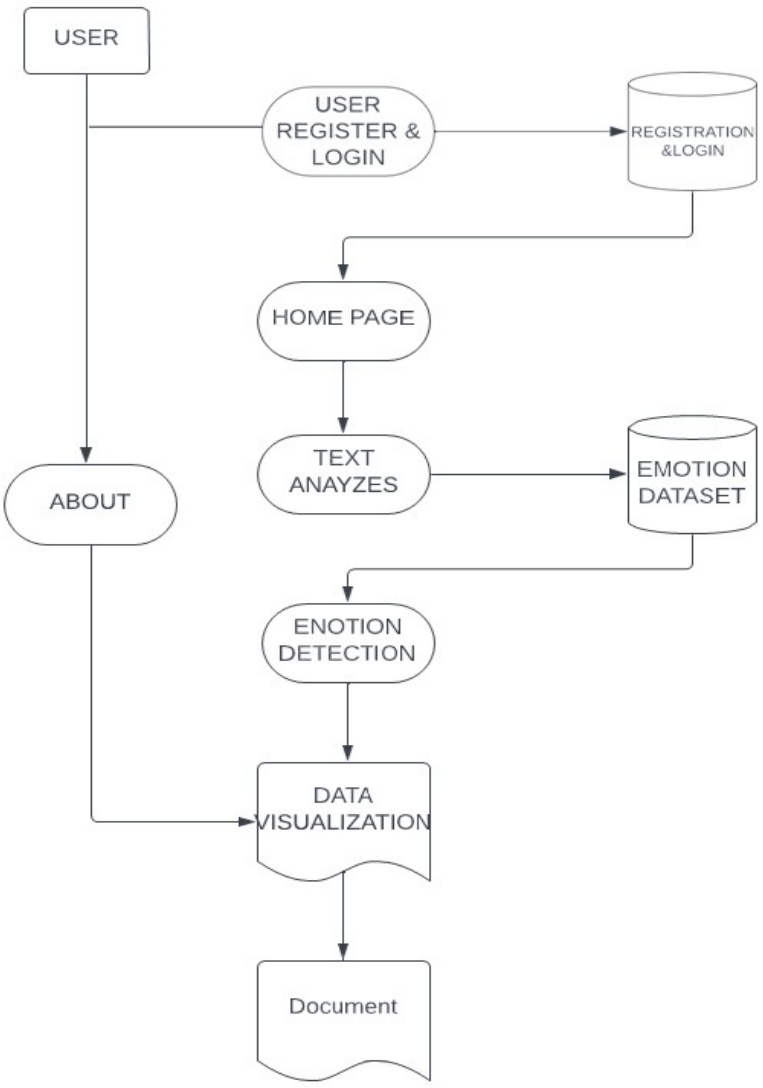


DATAFLOW DIAGRAM

LEVEL 0



LEVEL 1



5.4 DATABASE DESIGN

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. Database management system manages the data accordingly. Database design involves classifying data and identifying interrelationships.

5.4.1 DATABASE TABLE

FIELD NAME	DATA TYPE	SIZE	DESCRIPTION
User ID	INT	10	It is used to store the unique id
Username	Varchar	50	It is used to store the username
Password	Varchar	25	It is used to store the password
First_Name	Varchar	50	It is used to store the First_Name
Last_Name	Varchar	50	It is used to store the Lasst_name
Age	INT	10	It is used to store the Age
Gender	Varchar	10	It is used to store the Gender
Email	Varchar	100	It is used to store the Email

5.5 SYSTEM DESIGN

The system design of the "Emotion Detection in Text Data" project is a meticulously crafted framework that seamlessly integrates cutting-edge Natural Language Processing (NLP) techniques for precise emotion analysis in textual content. At its core, the system relies on a well-curated dataset, where each text sample is intricately paired with its corresponding expressed emotions, forming the bedrock for robust model training. The integration of core packages facilitates efficient data management, streamlined machine learning model loading, and ensures a robust database functionality, enhancing the system's overall performance.

User-centric features such as registration and login capabilities prioritize personalization, allowing users to input text data and promptly receive emotion predictions accompanied by confidence scores. This not only enhances user engagement but also provides transparency to the prediction process, instilling confidence in users and decision-makers relying on the system. Immediate feedback further augments user experience, creating a responsive interface that caters to the dynamic nature of emotion detection in textual data. The project goes beyond technical prowess by prioritizing accessibility and user-friendliness, aiming to demystify emotion detection in text. Its versatility extends across sectors, ranging from social media sentiment analysis to customer feedback appraisal and market research. The system's sophistication lies in its ability to decode intricate nuances of emotions conveyed through written language, offering a valuable tool for understanding and analyzing textual emotional content.

Scalability and adaptability are key considerations in the system design, achieved through the seamless integration of core packages that keep pace with evolving NLP technologies. User registration and login features not only enhance practicality but also contribute to the system's scalability by accommodating a growing user base. An administrative feature is strategically included to ensure smooth operation, placing a strong emphasis on compliance with user privacy and data security standards. The project's impact reverberates across industries, providing enhanced insights in social media sentiment analysis for accurate public opinion gauging and in-depth customer feedback evaluation for continuous product and service improvement. In the rapidly evolving landscape of technology, this project stands at the forefront, offering a sophisticated and comprehensive solution to decode and understand the intricate world of emotions within textual data, meeting the rising demand for effective emotion detection as technology advances and textual data volumes continue to grow.

5.6 INPUT DESIGN

The "Emotion Detection in Text Data" project is a sophisticated application designed to harness the capabilities of Natural Language Processing (NLP) for the meticulous identification and analysis of emotions embedded in textual content. Emotions, being intricate and nuanced, are decoded by a meticulously trained model, offering applications ranging from sentiment analysis and customer feedback evaluation to social media monitoring.

At its core, the project seamlessly integrates various packages for efficient data management, machine learning model loading, and database functionality. The model is trained on a carefully labeled dataset, where each text sample is paired with its corresponding expressed emotions. The user interface enhances personalization with features such as user registration and login, allowing users to input text data and receive immediate emotion predictions, complete with confidence scores.

The project's primary ambition is to demystify emotion detection in text, providing accessibility and user-friendliness. This capability extends its utility across diverse sectors, including social media sentiment analysis, customer feedback appraisal, market research, and beyond.

5.7 OUTPUT DESIGN

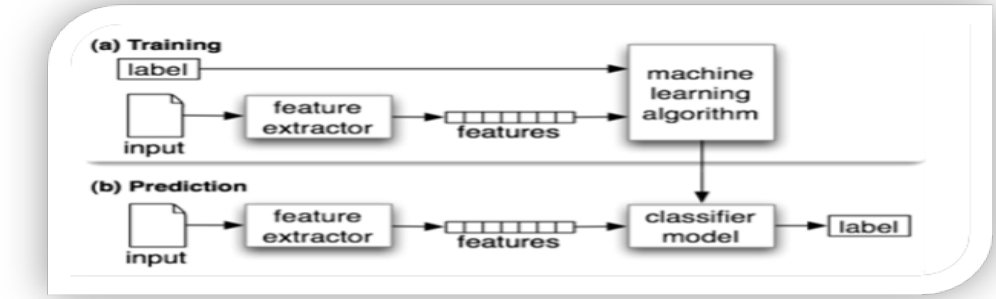
The "Emotion Detection in Text Data" project leverages Natural Language Processing (NLP) to intricately decipher emotions in textual content. Its core mission is to develop a highly accurate model for discerning emotions in text, serving applications such as sentiment analysis, customer feedback evaluation, and social media monitoring. The model's proficiency is honed through meticulous training on labeled datasets, pairing each text sample with its corresponding expressed emotions.The project seamlessly integrates essential packages for efficient data management, machine learning model loading, and database functionality. User registration and login features enhance personalization, allowing users to receive immediate emotion predictions and confidence scores upon inputting text. An administrative feature facilitates the monitoring of user activities for smooth operation.

The project aims to demystify emotion detection in text, offering accessibility and user-friendliness for diverse applications, including social media sentiment analysis and market research. The application provides real-time emotion detection, presenting users with instant analyses and confidence scores. The user-friendly interface caters to both data scientists and non-experts.

5.7 ALGORITHM

Natural Language Processing

NLP is a branch of data science that consists of systematic processes for analyzing, understanding, and deriving information from the text data in a smart and efficient manner. By utilizing NLP and its components, one can organize the massive chunks of text data, perform numerous automated tasks and solve a wide range of problems such as – automatic summarization, machine translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation etc.



Important Libraries for NLP (python)

- Scikit-learn: Machine learning in Python
- Natural Language Toolkit (NLTK): The complete toolkit for all NLP techniques.
- Pattern – A web mining module for the with tools for NLP and machine learning.
- TextBlob – Easy to use nl p tools API, built on top of NLTK and Pattern.
- spaCy – Industrial strength N LP with Python and Cython.
- Gensim – Topic Modelling for Humans
- Stanford Core NLP – NLP services and packages by Stanford NLP Group.

CHAPTER 6

SYSTEM TESTING

TESTING DESGRIPTION

After the source code has been completed, documented as related data structures. Completion of the project has to undergo testing and validation where there is subtitle and definite attempt to get errors. The project developer treats lightly, designing and execution of the project test that will demonstrates that the program works rather than uncovering errors, unfortunately errors will be present and if the project developer doesn't found errors, the user will find out.

The project developer is always responsible for testing the individual units i.e. modules of the program. In many cases, developer should conduct the integration testing i.e. the testing step that leads to the construction of the complete program structure..

6.1 UNIT TESTING

The provided code lacks explicit implementation or mention of unit testing. However, potential areas for unit testing include functions responsible for user authentication, registration, and emotion prediction, such as make_hashes, check_hashes, add_userdata, login_user, and predict_emotions. Unit tests can ensure that user authentication functions appropriately handle password hashing and validation, confirm that user registration functions correctly insert data into the database, and validate the accuracy of the emotion, including confidence scores. The implementation of unit tests in these critical areas contributes to code reliability, ensuring each function performs as expected and enhancing the overall robustness of the "Emotion Detection in Text Data" project.

6.2 INTEGRATION TESTING

The "Emotion Detection in Text Data" project employs rigorous integration testing to validate seamless collaboration among its components. This approach ensures the effective interaction of modules, packages, and functionalities within the application. Key integration points involve coordinating the NLP model, database management, user authentication, and

data visualization using Altair and Plotly Express. The inclusion of SQLite for database operations exemplifies a comprehensive testing strategy, verifying the cohesive functioning of these elements as a unified system. The testing process extends to user registration, login processes, and the smooth transition between application states, guaranteeing a reliable and robust user experience.

6.3 USER INTERFACE TESTING

The "Emotion Classifier App" employs a robust UI testing strategy using Streamlit to ensure a seamless user experience. Key components include rigorous testing of menu navigation for precise redirection, validation of user registration and login functionalities, scrutiny of real-time emotion detection for accurate display, and evaluation of administrative features. Monitoring page metrics and validating the about section's content contribute to the comprehensive UI testing. The registration form undergoes thorough checks, confirming successful registration and smooth redirection to the login page. Logout functionality is tested to ensure users are successfully logged out.

6.4 SECURITY TESTING

The project underscores security with robust measures: Passwords undergo secure SHA-256 hashing for heightened protection. User registration features stringent input validation, ensuring data integrity. st.session_state in Streamlit ensures secure session management, curbing unauthorized access. Database security employs parameterized SQL queries to ward off SQL injection risks. Machine learning model loading is done securely from a specified location, minimizing potential threats. Admin login follows stringent validation, curbing unauthorized access to sensitive features. The "Monitor" section, utilizing Streamlit's expander, enhances security by controlling content expansion, and averting information disclosure to unauthorized users.

6.5 TEST CASES

Test Case	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Registration	Users effortlessly create accounts by inputting vital details like names, a preferred login username, age, gender, and a Gmail-validated email address.	Registr ation sucess ful	Details are incorrect	Fail	The user register the page and gmail id is invalid
2	Login	A login form utilizes the credentials of a user, in order to authenticate their access	Login sucessful	Incorrect user name	Fail	The user enter the details to the login page the password and username are incorrect
3	Emotion Detection	Users input text into a dedicated area, triggering the pre-trained machine learning pipeline to predict emotions associated with the provided content.	Text converte d to the emoji	Emoji are converte d	Pass	The user enter the emotional text and that test converted into the emoji
4	Monitor	With a focus on enhancing user experience and performance monitoring, the Monitor section displays page metrics, including the frequency of visits to different sections of the application.	Admin moniter the website	View the website	Pass	ensuring that administrator s can track user activities seamlessly for continuous optimization.

CHAPTER 7
SYSTEM IMPLEMENTATION

When The "Emotion Detection in Text Data" project is an advanced application leveraging Natural Language Processing (NLP) to precisely identify and analyze emotions within textual content. The project's primary objective is to develop a model proficient in discerning emotions, applicable to various purposes such as sentiment analysis, customer feedback evaluation, and social media monitoring. The implementation seamlessly integrates essential packages for data management, machine learning model loading, and database functionality. It features user registration and login capabilities to enhance personalization, allowing users to input text data and receive immediate emotion predictions with associated confidence scores. An administrative feature is included for monitoring user activities, ensuring smooth operation. The project aims to demystify emotion detection in text, providing accessibility and user-friendliness for applications across sectors, including social media sentiment analysis, customer feedback appraisal, and market research.

CORE PACKAGES AND COMPONENTS

The implementation utilizes core packages such as `track_utils`, `sqlite3`, `streamlet`, `Altair`, and `plotly. express`. It employs an emotion classifier model loaded from a pre-trained joblib file. The application includes functions for user registration, login, and data validation. Additionally, it incorporates a monitoring feature for tracking user activities. The main application is organized into sections, including "Home," "Registration," "Login," "About," "Logout," "Admin," and "Monitor," each catering to specific functionalities. The admin section provides an interface for admin login and user data monitoring.

CHAPTER 8
CONCLUSION & FUTURE ENHANCEMENT

8.1 CONCLUSION

The "Emotion Detection in Text Data" project is a comprehensive application leveraging Natural Language Processing (NLP) to discern and analyze emotions within textual content. The primary objective is to develop a model proficient in accurately identifying emotions, catering to diverse applications such as sentiment analysis, customer feedback evaluation, and social media monitoring. The project incorporates core packages for efficient data management, machine learning model loading, and database functionality. It features user registration and login functionalities, enhancing personalization. Users can input text data and receive immediate emotion predictions with associated confidence scores. The application also includes an administrative feature for monitoring user activities. Overall, the project aims to demystify emotion detection in text, offering accessibility and user-friendliness for applications in various sectors. The provided Python code outlines the implementation details, including model loading, database management, and user authentication, making it a versatile and powerful tool for understanding emotions in textual data.

8.2 FUTURE ENHANCEMENTS

The "Emotion Detection in Text Data" project exhibits a robust platform for emotion analysis in textual content. To advance its capabilities, key future enhancements include integrating multi-language support for broader usability, implementing real-time data streaming for continuous social media and customer feedback analysis, and fine-tuning the model on domain-specific datasets to enhance accuracy in industry contexts. The addition of advanced visualization techniques and sentiment trend analysis offers a deeper understanding of emotional patterns, while integration with external APIs enables sentiment comparison against global benchmarks. Strengthening user authentication with measures like two-factor authentication enhances security. Finally, the creation of an interactive dashboard enriches the user experience, solidifying the application as a versatile and cutting-edge tool for emotion analysis across diverse domains.

CHAPTER 9

APPENDIX

9.1 SOURCE CODE

```
from track_utils import create_page_visited_table, add_page_visited_details,
view_all_page_visited_details, add_prediction_details, view_all_prediction_details,
create_emotionclf_table
import hashlib
import sqlite3
import streamlit as st
import altair as alt
import plotly.express as px
import pandas as pd
import numpy as np
from datetime import datetime
import joblib
pipe_lr = joblib.load(open("./models/emotion_classifier_pipe_lr.pkl", "rb"))
def predict_emotions(docx):
    results = pipe_lr.predict([docx])
    return results[0]
def get_prediction_proba(docx):
    results = pipe_lr.predict_proba([docx])
    return results

emotions_emoji_dict = {
    "anger": "□", "disgust": "□", "fear": "□□", "happy": "□",
    "joy": "□", "neutral": "□", "sad": "□", "sadness": "□", "shame": "□", "surprise": "□"
}
conn = sqlite3.connect('data.db')
c = conn.cursor()
def make_hashes(password):
    return hashlib.sha256(str.encode(password)).hexdigest()
def check_hashes(password, hashed_text):
    if make_hashes(password) == hashed_text:
        return hashed_text
    return False
def create_usertable():
    c.execute('CREATE TABLE IF NOT EXISTS registration_table(username TEXT,
password TEXT, first_name TEXT, last_name TEXT, age INT, gender TEXT, email
```

```
TEXT)')
create_usertable()
def add_userdata(username, password, first_name, last_name, age, gender, email):
    c.execute('INSERT INTO registration_table(username, password, first_name,
last_name, age, gender, email) VALUES (?,?,?,?,?,?,?)',
              (username, password, first_name, last_name, age, gender, email))
    conn.commit()
def login_user(username, password):
    c.execute('SELECT * FROM registration_table WHERE username = ? AND password
= ?',
              (username, password))
    data = c.fetchall()
    return data
def view_all_users():
    c.execute('SELECT * FROM registration_table')
    data = c.fetchall()
    return data
def check_user_table():
    c.execute('SELECT * FROM registration_table')
    data = c.fetchall()
    return data
def is_valid_email(email):
    return email.endswith('@gmail.com')
def is_valid_name(name):
    return name.isalpha()
def is_valid_password(password):
    return len(password) == 8
def main():
    st.title("EMOTION CLASSIFIER APP")
    menu = ["Home", "Registration", "Login", "About", "Logout", "Admin", "Monitor"]
    choice = st.sidebar.selectbox("Menu", menu)
    create_page_visited_table()
    create_emotionclf_table()
    st.session_state.admin_logged_in = False
    logged_in = False
    username = ""
    if choice == "Login" and not logged_in and not st.session_state.admin_logged_in:
        st.subheader("Login Section")
        username = st.sidebar.text_input("User Name")
        password = st.sidebar.text_input("Password", type='password')
```

```
        if st.sidebar.checkbox("Login"):
            create_usertable()
            if not is_valid_password(password):
                st.error("Invalid password format. Password should be exactly 8 characters
long.")
            else:
                hashed_pswd = make_hashes(password)
                result = login_user(username, check_hashes(password, hashed_pswd))
                if result:
                    st.success("Logged In as {}".format(username))
                    logged_in = True
                    choice = "Home"
                else:
                    st.error("Invalid credentials. Please try again.")
                    st.write("User Table Data:", check_user_table())

    if not logged_in and not st.session_state.admin_logged_in:
        if choice == "Admin":
            admin_username = "admin"
            admin_password = "admin"
            st.subheader("Admin Login Section")
            username = st.text_input("Admin Name")
            password = st.text_input("Admin Password", type='password')
            submit = st.button("Submit")
            cancel = st.button("Cancel")
            if submit:
                if username == admin_username and password == admin_password:
                    st.session_state.admin_logged_in = True
                    st.success("Admin Login Successful")
                    choice = "View User Data"  # Redirect to the "View User Data" page
                else:
                    st.error("Invalid admin credentials. Please try again.")
            if cancel:
                choice = "Home"
        if choice == "Monitor":
            add_page_visited_details("Monitor", datetime.now())
            st.subheader("Monitor App")
            with st.expander("Page Metrics"):
                page_visited_details = pd.DataFrame(view_all_page_visited_details(),
columns=['Page Name', 'Time of Visit'])
```

```
            st.dataframe(page_visited_details)
            pg_count = page_visited_details['Page
Name'].value_counts().rename_axis('Page Name').reset_index(name='Counts')
            c = alt.Chart(pg_count).mark_bar().encode(
               x='Page Name', y='Counts', color='Page Name')
            st.altair_chart(c, use_container_width=True)
            p = px.pie(pg_count, values='Counts', names='Page Name')
            st.plotly_chart(p, use_container_width=True)
    if choice == "About":
        add_page_visited_details("About", datetime.now())
        st.write("Welcome to the Emotion Detection in Text App! This application utilizes
the power of natural language processing and machine learning to analyze and identify
emotions in textual data.")
        st.subheader("Our Mission")
        st.write("At Emotion Detection in Text, our mission is to provide a user-friendly and
efficient tool that helps individuals and organizations understand the emotional content
hidden within text. We believe that emotions play a crucial role in communication, and by
uncovering these emotions, we can gain valuable insights into the underlying sentiments
and attitudes expressed in written text.")
        st.subheader("How It Works")
        st.write("When you input text into the app, our system processes it and applies
advanced natural language processing algorithms to extract meaningful features from the
text. These features are then fed into the trained model, which predicts the emotions
associated with the input text. The app displays the detected emotions, along with a
confidence score, providing you with valuable insights into the emotional content of your
text.")
        st.subheader("Key Features:")
        st.markdown("##### 1. Real-time Emotion Detection")
        st.write("Our app offers real-time emotion detection, allowing you to instantly
analyze the emotions expressed in any given text. Whether you're analyzing customer
feedback, social media posts, or any other form of text, our app provides you with
immediate insights into the emotions underlying the text.")
        st.markdown("##### 2. Confidence Score")
        st.write("Alongside the detected emotions, our app provides a confidence score,
indicating the model's certainty in its predictions. This score helps you gauge the
reliability of the emotion detection results and make more informed decisions based on the
analysis.")
        st.markdown("##### 3. User-friendly Interface")
        st.write("We've designed our app with simplicity and usability in mind. The intuitive
user interface allows you to effortlessly input text, view the results, and interpret the
emotions detected. Whether you're a seasoned data scientist or someone with limited
technical expertise, our app is accessible to all.")
        st.subheader("Applications")
        st.markdown("The Emotion Detection in Text App has a wide range of applications
```

```
across various industries and domains. Some common use cases include:")
        st.write("- Social media sentiment analysis")
        st.write("- Customer feedback analysis")
        st.write("- Market research and consumer insights")
        st.write("- Brand monitoring and reputation management")
        st.write("- Content analysis and recommendation systems")
    if logged_in and choice == "Home":
        add_page_visited_details("Home", datetime.now())
        st.subheader("Emotion Detection in Text data")
        with st.form(key='emotion_clf_form'):
            raw_text = st.text_area("Type Here")
            submit_text = st.form_submit_button(label='Submit')
        if submit_text:
            col1, col2 = st.columns(2)
            prediction = predict_emotions(raw_text)
            probability = get_prediction_proba(raw_text)
            add_prediction_details(raw_text, prediction, np.max(
                probability), datetime.now())
            with col1:
                st.success("Original Text")
                st.write(raw_text)
                st.success("Prediction")
                emoji_icon = emotions_emoji_dict[prediction]
                st.write("{}:{}".format(prediction, emoji_icon))
                st.write("Confidence: {}".format(np.max(probability)))
            st.success("Prediction Probability")
            proba_df = pd.DataFrame(probability, columns=pipe_lr.classes_)
            proba_df_clean = proba_df.T.reset_index()
            proba_df_clean.columns = ["emotions", "probability"]
            fig = alt.Chart(proba_df_clean).mark_bar().encode(
                x='emotions', y='probability', color='emotions')
            st.altair_chart(fig, use_container_width=True)
    elif choice == "Logout":
        logged_in = False
        st.success("Logged Out")
        st.info("You are now logged out. Please log in again to access the application.")
    if choice == "Registration":
        st.title("Registration Form")
        first_name = st.text_input("First Name")
        last_name = st.text_input("Last Name")
```

```
        reg_username = st.text_input("Username (Use for Login)")
        age = st.number_input("Age", min_value=0, max_value=120, step=1)
        gender = st.radio("Gender", ("Male", "Female", "Other"))
        email = st.text_input("Email (Gmail)")
        reg_password = st.text_input("Password (Use for Login)", type="password")
        confirm_password = st.text_input("Confirm Password", type="password")
        if st.button("Register"):
            if not is_valid_email(email):
                st.error("Invalid email format. Please use a Gmail address.")
            elif not is_valid_name(first_name):
                st.error("Invalid first name. Use only alphabets.")
            elif not is_valid_name(last_name):
                st.error("Invalid last name. Use only alphabets.")
            elif len(reg_password) != 8:
                st.error("Password must be exactly 8 characters long.")
            elif reg_password != confirm_password:
                st.error("Passwords do not match. Please re-enter.")
            else:
                create_usertable()  # Create the registration table
                hashed_pswd = make_hashes(reg_password)
                add_userdata(reg_username, hashed_pswd, first_name, last_name, age, gender,
email)
                st.success("Registration successful! You can now log in.")
                choice = "Login"  # Redirect to the login page after successful registration
    if st.session_state.admin_logged_in:
        if choice == "View User Data":
            add_page_visited_details("View User Data", datetime.now())
            st.subheader("View User Registration Data")
            user_data = view_all_users()
            if user_data:
                user_df = pd.DataFrame(user_data, columns=["Username", "Password", "First
Name", "Last Name", "Age", "Gender", "Email"])
                st.dataframe(user_df)
            else:
                st.warning("No user registration data found in the database.")
if __name__ == '__main__':
    main()
```

9.2 SCREENSHOTS

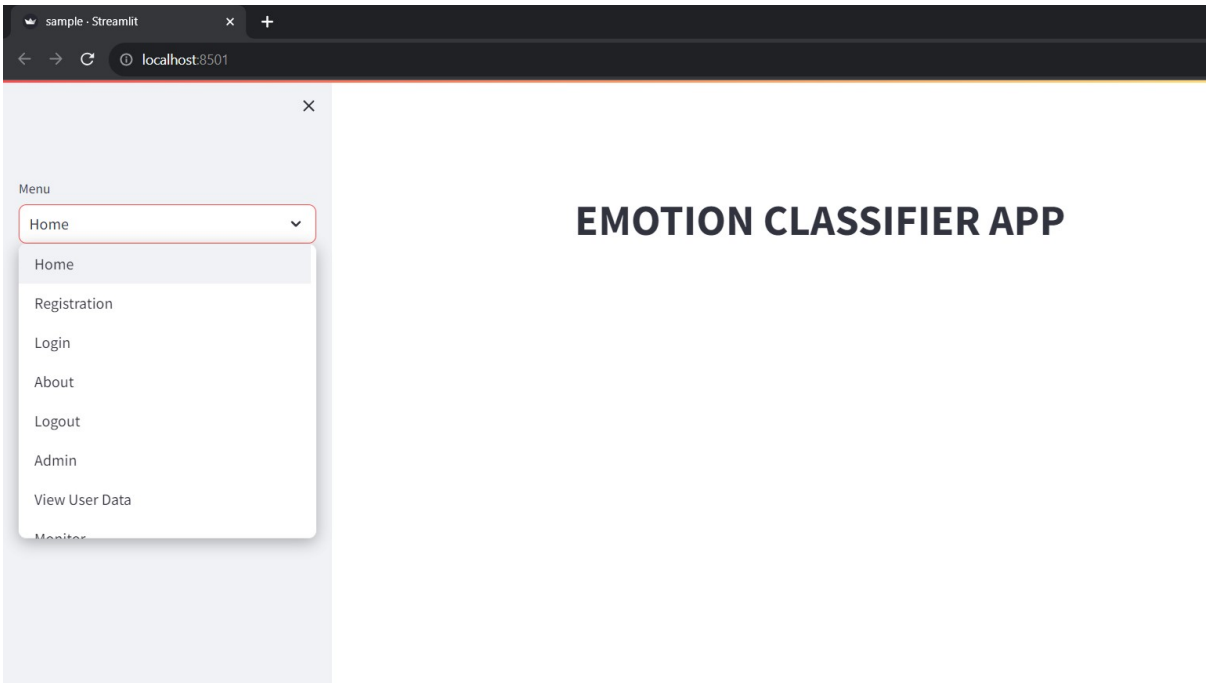


Fig 9.2.1.Home Page

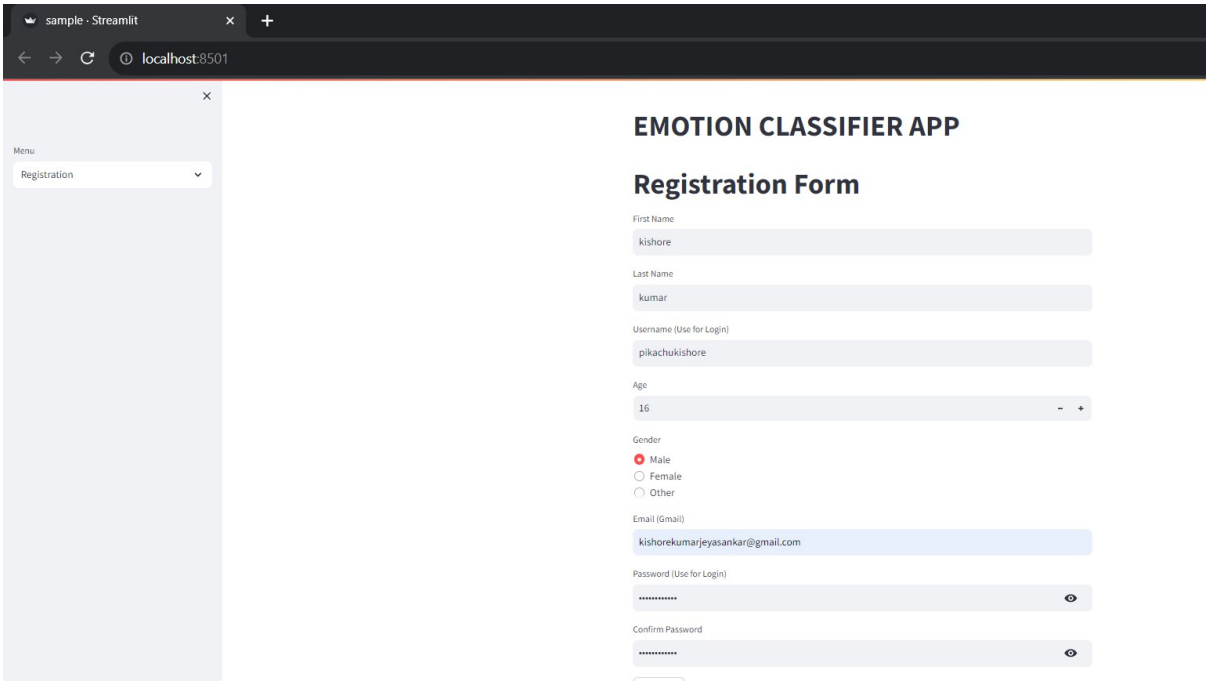


Fig 9.2.2.Registration Page

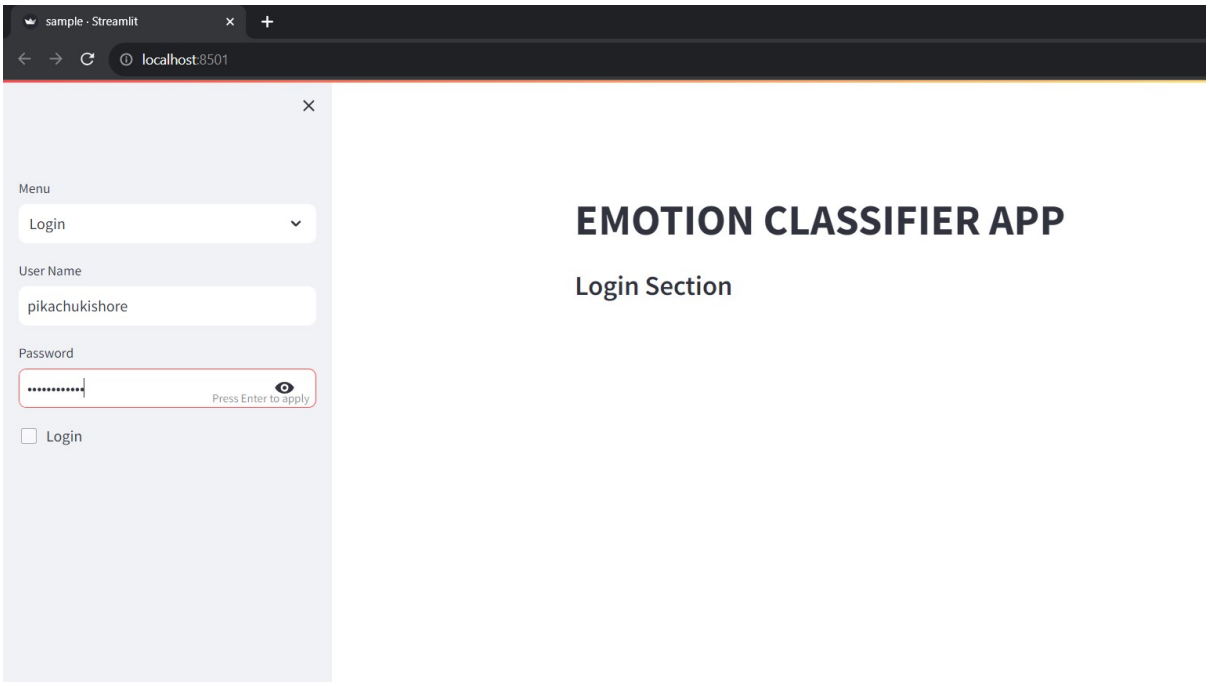


Fig 9.2.3.Login Page

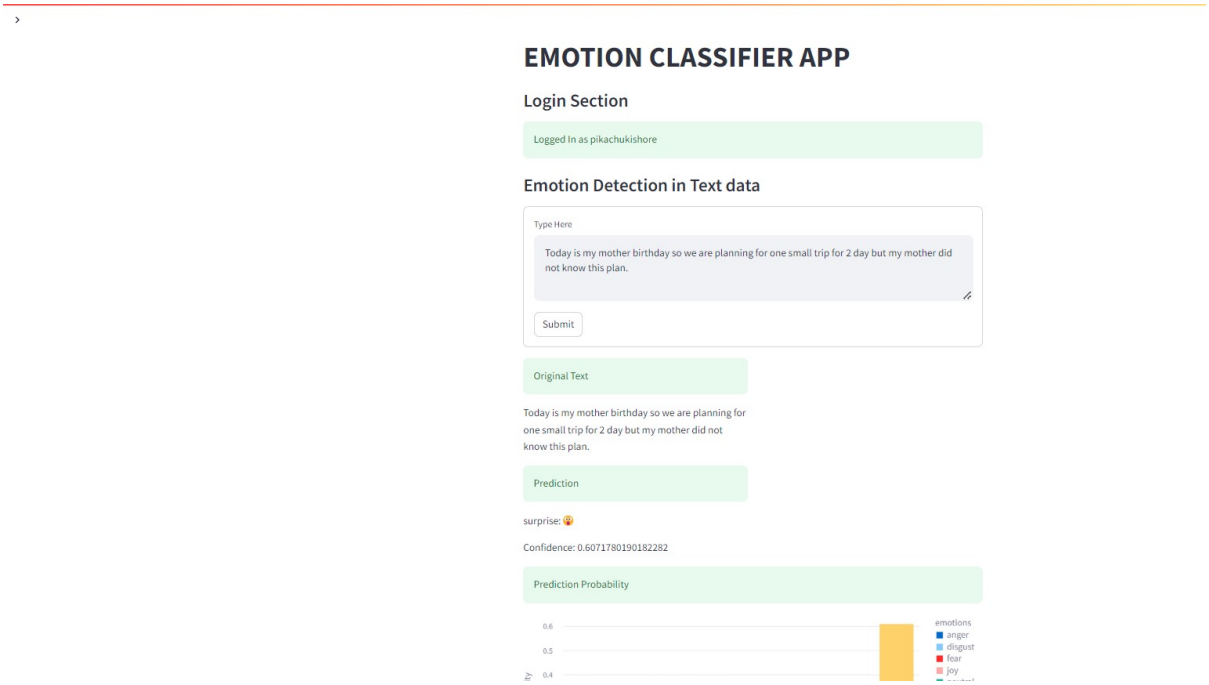


Fig 9.2.4.Emotion Detection

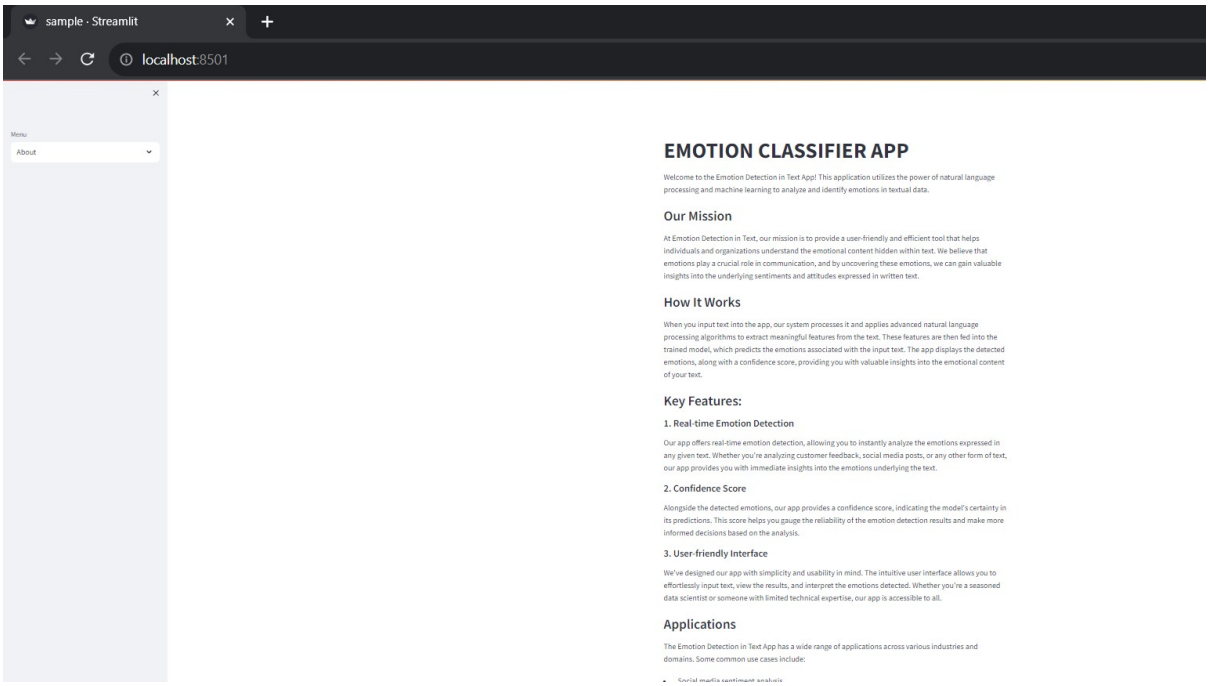


Fig 9.2.5.About

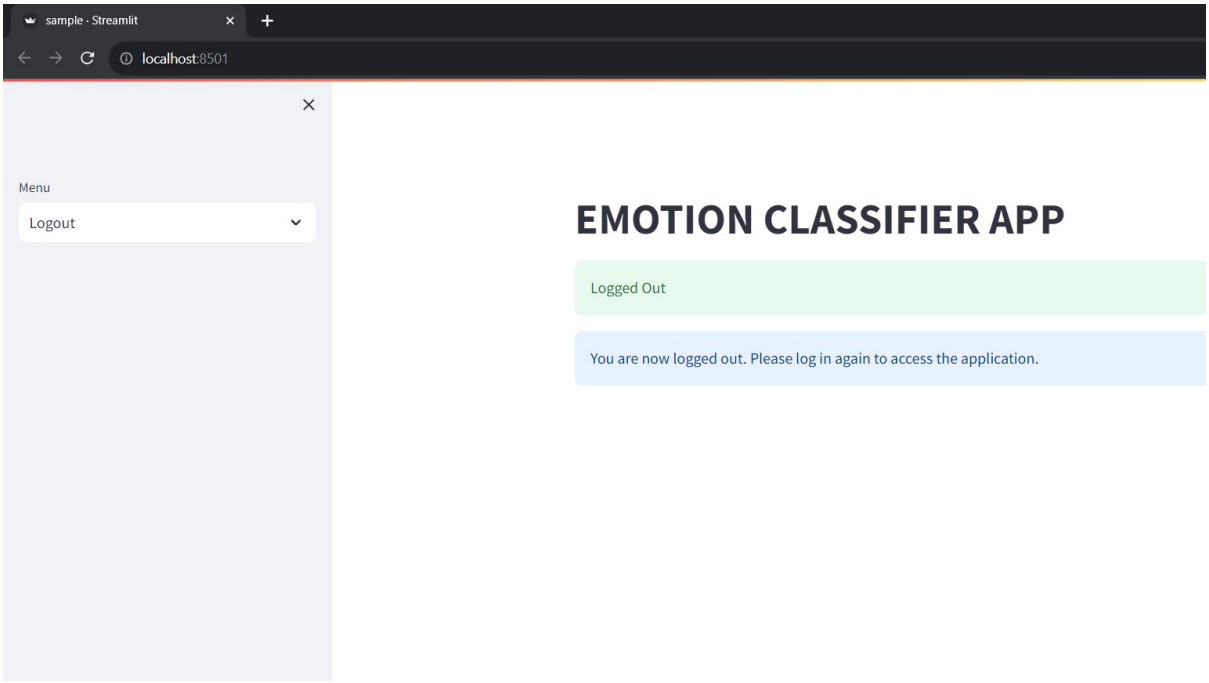


Fig 9.2.6.Logout

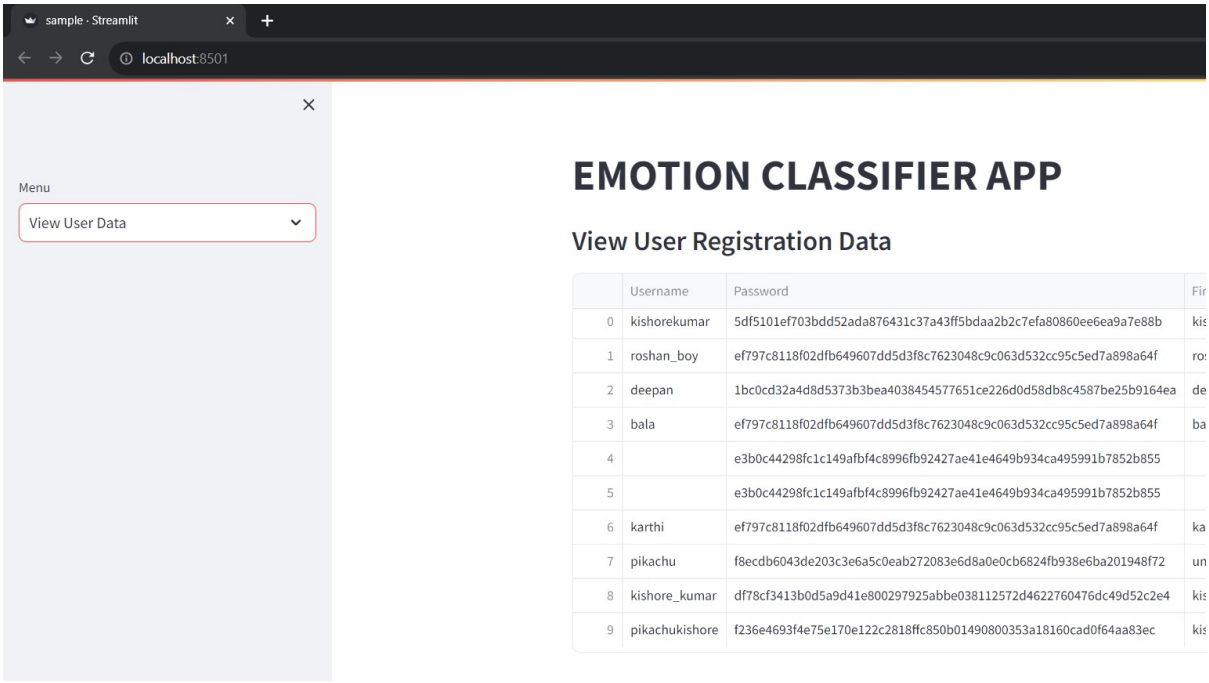


Fig 9.2.7.Admin

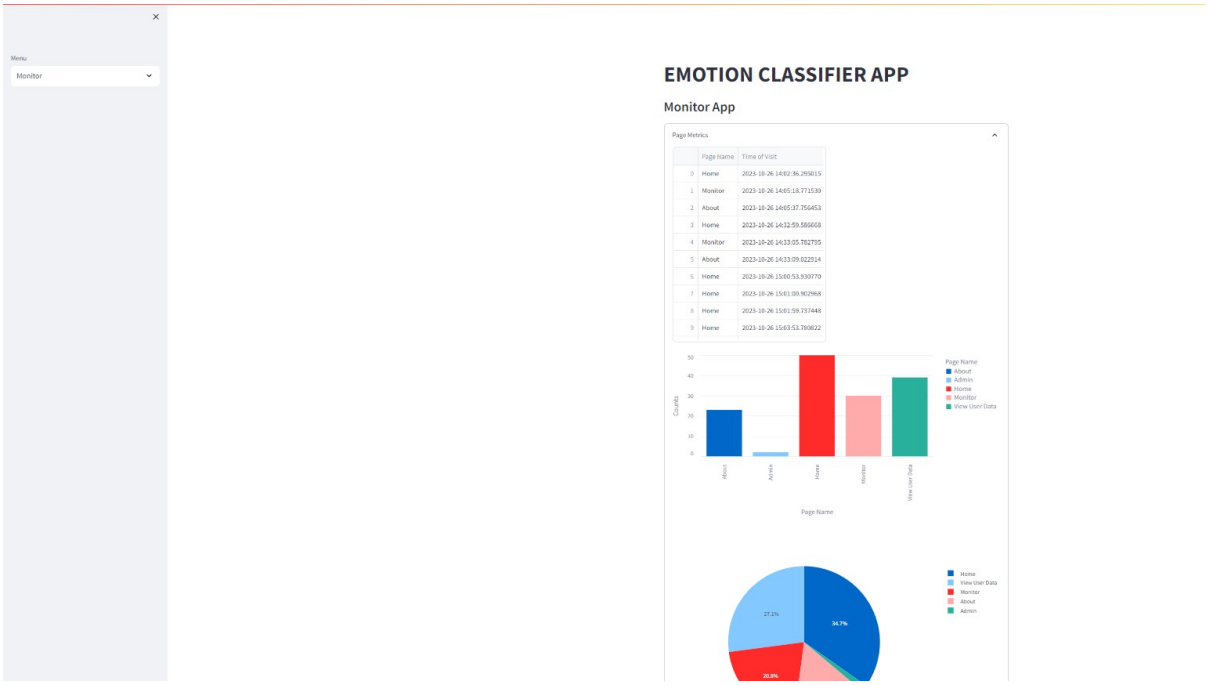


Fig 9.2.8.Monitor

CHAPTER 10
REFERENCES

10.1 BOOK REFERENCES

[1] "Natural Language Processing in Action" by Lane, Howard, and Hapke

[2] "Speech and Language Processing" by Jurafsky & Martin

[3] "Applied Natural Language Processing with Python" by Lane, Gauthier, and Howard

[4] "Foundations of Statistical Natural Language Processing" by Manning and Schütze

[5] "Emotion AI: The Intersection of Emotion and Artificial Intelligence" by Rana el Kaliouby

[6] "Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data" by Dipanjan Sarkar

[7] "Python Natural Language Processing" by Jalaj Thanaki

[8] "Mastering Natural Language Processing with Python" by Deepti Chopra, Nisheeth Joshi, and Iti Mathur

10.2 REFERRED WEBSITES

- ✓ https://link.springer.com/article/10.1007/s13278-021-00776-6
- ✓ https://www.python.org/
- ✓ https://www.w3schools.com/
- ✓ https://www.javatpoint.com/types-of-machine-learning
- ✓ https://www.javatpoint.com/natural-language-toolkit.com/
- ✓ https://www.latentview.com/data-engineering-lp/introduction-to-streamlit/#:~:text=Streamlit%20is%20an%20open%20source,NumPy%2C%20pandas%2C%20Matplotlib%20etc.
- ✓ https://www.datacamp.com/tutorial/streamlit
- ✓ https://docs.python.org/3/library/sqlite3.html#:~:text=Source%20code%3A%20Lib%2Fsqlite3%2F,SQLite%20for%20internal%20data%20storage.