EMOTION DETECTION IN TEXT DATA USING NLP

Submitted by

KISHORE KUMAR J (711522MMC022)

Of

KIT – KALAIGNARKARUNANIDHI INSTITUTE OF TECHNOLOGY
(An Autonomous Institution)

OPEN LABORATORY PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION

ENGINEERING

In partial fulfilment of award of the

degree of

MASTER OF COMPUTER APPLICATIONS

ANNA UNIVERSITY : CHENNAI – 600 025

DEC – 2023

BONAFIDE CERTIFICATE

Certified that this open laboratory project report **"EMOTION DETECTION IN TEXT DATA USING NLP"** is the bonafide work of **Mr. KISHORE KUMAR J (711522MMC022)** who carried out the project work under my supervision.

Ms. R. SUGANYA MCA.,	**Dr. E. Vijayakumar MCA., Ph.D.,**
Associate Professor Project Guide,	Associate Professor & Head,
Master of Computer Applications,	Master of Computer Applications,
KIT – Kalaignarkarunanidhi Institute of Technology,	KIT – Kalaignarkarunanidhi Institute of Technology,
Kannampalayam,	Kannampalayam,
Coimbatore (Dt.).	Coimbatore (Dt.).

Submitted to the Anna University Viva-Voce held on____________________

Internal Examiner **External Examiner**

DECLARATION

I affirm that the project work titled **"EMOTION DETECTION IN TEXT DATA USING NLP"** beingsubmitted in partial for the award of MCA is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

(Signature of the Candidate)

KISHORE KUMAR J

711522MMC022

I certify that the declaration made above by the candidate is true

Signature of the Guide,

Ms. R. Suganya MCA.,

Assistant Professor.

ACKNOWLEDGEMENT

My heartfelt gratitude and thanks to the Almighty God, my parents and other family members and friends for providing the opportunity to undergo this project successfully in this esteemed institution.

At the outset, I would like to thank our Founder and Chairman **Thiru. PONGALUR N. PALANISAMY, KIT - Kalaignarkarunanidhi Institute of Technology,** who has given me an opportunity to undergo this Project Work, successfully in this esteemed Institution.

I express my sincere thanks to **Mrs. P. INDU MURUGESAN, Vice Chairperson, KIT - Kalaignarkarunanidhi Institute of Technology,** who encouraged me by giving her support and constant encouragement.

I extend my grateful thanks and wishes to **Dr. N. MOHANDAS GANDHI, ME., MBA. Ph.D., CEO, KIT - Kalaignarkarunanidhi Institute of Technology,** for the valuable suggestion in framing my carrier towards the fulfillment of this Project work.

I express my sincere thanks to **Dr. M. RAMESH, ME., Ph.D., Principal, KIT-Kalaignarkarunanidhi Institute of Technology,** who encouraged me by giving his valuable suggestion and constant encouragement.

I would like to acknowledge the respective **Dr. E. VIJAYAKUMAR MCA., Ph.D., Associate and Head Department of Computer Applications, KIT - Kalaignarkarunanidhi Institute of Technology,** for spending the valuable time in guiding and supporting me to make this project a successful one.

I take the privilege to extend my hearty thanks to our project Coordinator and my internal guide **Ms. R. SUGANYA M.C.A., Assistant professor, Department of Computer Applications** for spending his valuable time and energy in guiding, supporting and helping me in preparation of the project.

Finally with great enthusiasm I express my thanks to all the faculty members for providing necessary information and their sustained interest in my part of successful completion.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1.1 OBJECTIVE

The "Emotion Detection in Text Data" project is a sophisticated application that harnesses the power of Natural Language Processing (NLP) to meticulously identify and scrutinize emotions embedded in textual content. Emotions are intricate, and this project strives to decode them, acknowledging their subtle nuances conveyed through the written word.

Its primary mission is to craft a model adept at precisely discerning emotions within text data. This model serves a broad spectrum of purposes, ranging from sentiment analysis and customer feedback evaluation to social media monitoring. Its proficiency is honed through training on a meticulously labeled dataset, where each text sample is paired with its corresponding expressed emotions.

The project seamlessly integrates an array of core packages for efficient data management, loading machine learning models, and seamless database functionality. It further incorporates user registration and login features, enhancing personalization for users. Upon inputting text data, users can expect immediate emotion predictions from the model, along with associated confidence scores. Additionally, an administrative feature empowers the monitoring of user activities, ensuring smooth operation.

In essence, this project's core ambition is to demystify emotion detection in text, offering accessibility and user-friendliness. This capability lends itself to diverse applications across sectors, including social media sentiment analysis, customer feedback appraisal, market research, and beyond.

CHAPTER 2
SYSTEM STUDY

2.1EXISTING 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.

CHAPTER 3
SYSTEM SPECIFICATION

HARDWARE CONFIGURATION

This section gives the details and specification of the hardware on which the system is expected to work.

Processor	: Dual-core processor or higher.
RAM	: 4 GB RAM .
Hard Disk	: At least 10 GB of free disk space.
Keyboard	: Standard keyboard.

SOFTWARE SPECIFICATION

This section gives the details of the software that are used for the development.

Environment	: Visual Studio Code.
Language	: Python Streamlit
Application	: Web Application.
Operating System	: Windows®10, Mac OS*, and Linux.

CHAPTER 4
SOFTWARE DESCRIPTION

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. Many modern data-heavy apps face the challenge of building an effective user interface quickly, without taking complicated steps. Streamlit is a promising open-source Python library, which enables developers to build attractive user interfaces in no time.

FEATURES OF STREAMLIT

- **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.

BACKEND

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.

FEATURES OF SQLite

1. **SQLite is serverless:** SQLite doesn't require a different server process or system to operate.
2. **SQLite is very flexible:** It facilitates you to work on multiple databases in the same session at the same time.
3. **Configuration Not Required:** SQLite doesn't require configuration. No setup or administration is required.
4. **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.
5. **Storing data is easy:** SQLite provides an efficient way to store data.
6. **Provide a large number of APIs:** SQLite provides API for a large range of programming languages.

CHAPTER 5
PROJECT DESCRIPTION

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.

MODULES

The system is computerized Web based Budget Tracking System. The following are the modules in the project

- **HOME**
- **REGISTRATION**
- **LOGIN**
- **EMOTION DETECTION**
- **ABOUT**
- **LOGOUT**
- **ADMIN**
- **MONITOR**

5.2.1 MODULE 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. In essence, the home page delivers real-time emotion detection with a seamless and user-friendly design, providing users with immediate insights into the emotional nuances of their input text.

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."

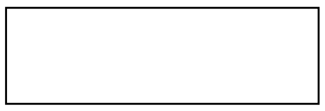
MONITOR

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.

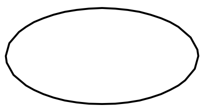
DATA FLOW DIAGRAM

The Data Flow Diagram is a graphical representation which depicts the information regarding the flow of control and the transformation of the data from the input to the output. The DFD may be used to represent the system or software at any level of abstraction. In fact DFD's may be partitioned into levels. A Level 0 DFD, also called a context model, represents the entire software element as a single bubble with input and output arrows. DFD is a graphical tool used for requirement analysis. DFD depicts information flow without explicit representation procedural logic. DFD also depicts real time systems with the help of some extended notation.

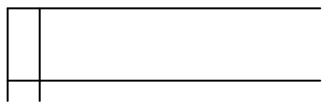
Basic DFD Notations



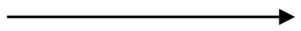
- Source (or) Destination



- Process

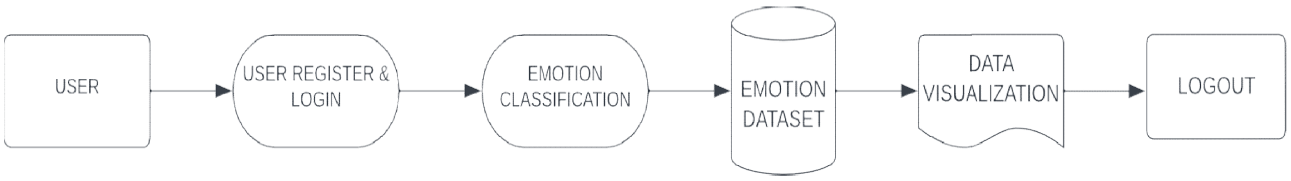


- Data Storage

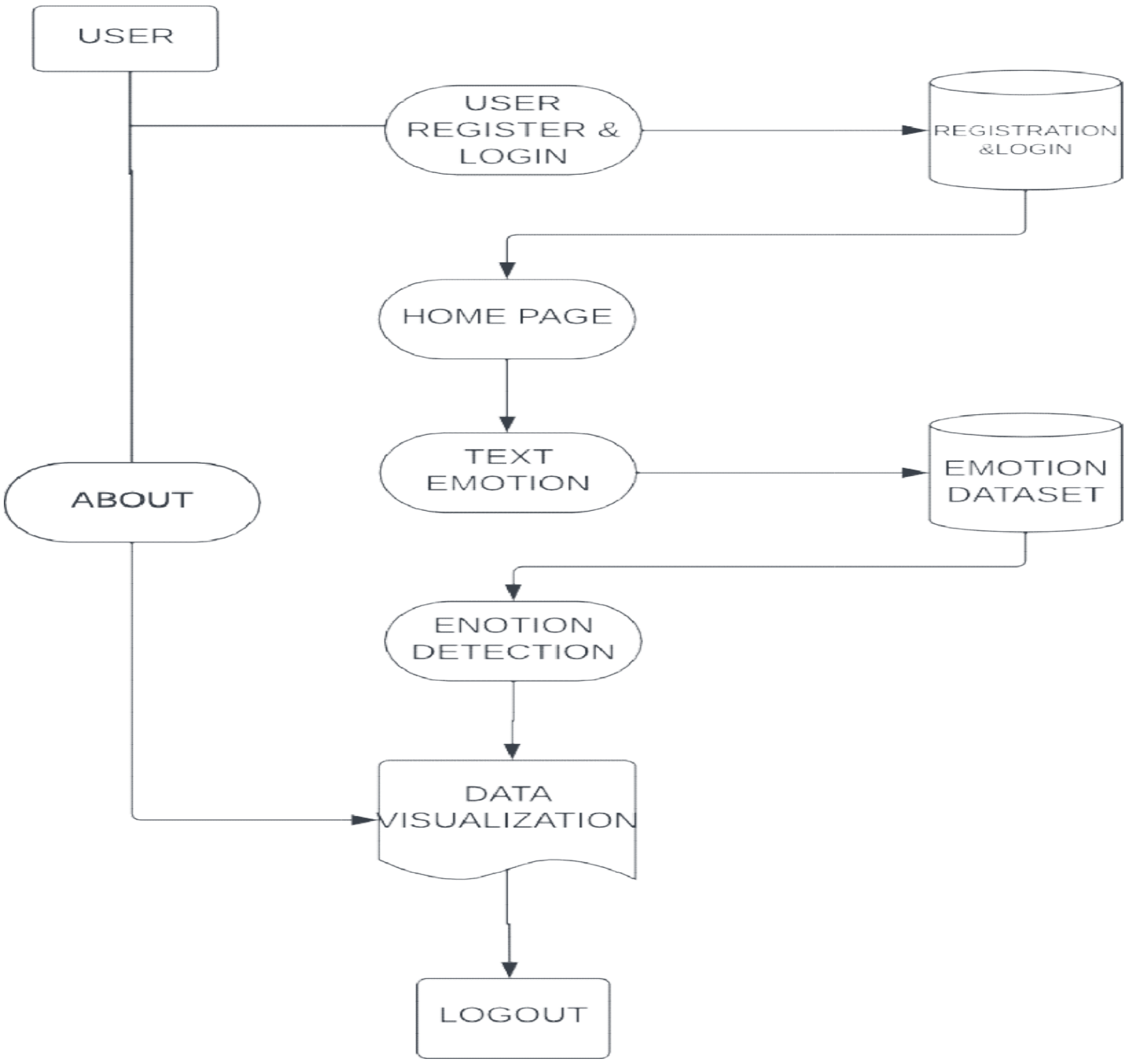


- Data Flow

Level - 0



Level - 1



DATABASE DESIGN (TABLE DESIGN)

A database design is a collection of inter-related data stored with a minimum of redundancy to serve many applications”. To develop a database that satisfies the information needs of today as well as for tomorrow, it is necessary to understand the database conceptually. The first task of the designer is to develop the conceptual model. The steps in the development of the conceptual model are

- ➢ Data Analysis
- ➢ Relational Identity
- ➢ Graphical Representation
- ➢ Design Process

The database design is an important factor both in system testing and maintaining. It minimizes the artificial embedded in using separate files. The primary objective of the database design is fast response in less time. The database has three major aspects.

STRUCTURES

Structures are well-defined object that store the data of the database.

OPERATORS

The integrity rule is a law that governs which operation are allowed on the data structures of a database.

TABLE DESIGN

TABLE NAME : registration_tablePrimary

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
User ID	INT	10	Primary key
Username	Varchar	50	Unique, Not Null
Password	Varchar	25	Not Null
First_Name	Varchar	50	Not Null
Last_Name	Varchar	50	Not Null
Age	INT	10	Not Null
Gender	Varchar	10	Not Null
Email	Varchar	100	Not Null

TABLE NAME : prediction_details_table

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
Text	TEXT	10	Not Null
Emotiom	TEXT	10	Not Null
Confidence	REAL	50	Not Null
Timestamp	DATETIIME	25	Not Null

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.

The code snippet presented showcases the implementation of the project in Python, employing core packages for data manipulation, model loading, and database operations. The main application encompasses a user-friendly interface for emotion detection, real-time prediction, and display of confidence scores. Additionally, the code includes a registration system, validation functions, and an admin module for viewing user data.

The Emotion Detection in Text Data project combines advanced NLP techniques with a robust user interface to offer a powerful tool for understanding and analyzing emotions in textual data, making it valuable across various industries and applications.

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.

The app finds applications in social media sentiment analysis, customer feedback evaluation, market research, brand monitoring, and content analysis. The "About" section emphasizes the project's mission, working principles, and key features in uncovering valuable insights from written text.

In conclusion, the "Emotion Detection in Text" project is a comprehensive and user-centric solution, contributing significantly to the evolving landscape of NLP applications.

CHAPTER 6
SYSTEM TESTING

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.

This project has undergone the following testing procedures to ensure its correctness.

- UNIT TESTING
- INTEGRATION TESTING
- USER INTERFACE TESTING
- SECURITY TESTING

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.

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

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.

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.

CHAPTER 7
SYSTEM IMPLEMENTATION

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.

USER REGISTRATION AND LOGIN

The application facilitates user registration with a form for inputting user details, ensuring email and password validity. User login is secured through hashed passwords and successful authentication grants access to the home page for emotion detection.

EMOTION DETECTION AND PREDICTION

The core functionality lies in the home page, where users can input text for real-time emotion detection. The application predicts emotions, displays the original text, associates emotions with emoji icons, and provides a confidence score. The results include a visualization of prediction probabilities for different emotions.

ABOUT AND APPLICATIONS

The "About" section explains the project's mission, working mechanism, key features, and applications. It emphasizes real-time emotion detection, confidence scores, and a user-friendly interface. The application's wide-ranging applications across social media sentiment analysis, customer feedback analysis, market research, brand monitoring, and content analysis are highlighted.

ADMIN AND MONITORING

The application features an admin section with login capabilities, providing access to view user registration data. The monitoring section tracks user activities, displaying page metrics, and counts of visited pages, enhancing transparency and accountability.

CHAPTER 8
CONCLUSION & FUTURE ENHANCEMENT

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 SCOPE FOR FUTURE ENHANCEMENT

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

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()
```

SCREENSHOTS

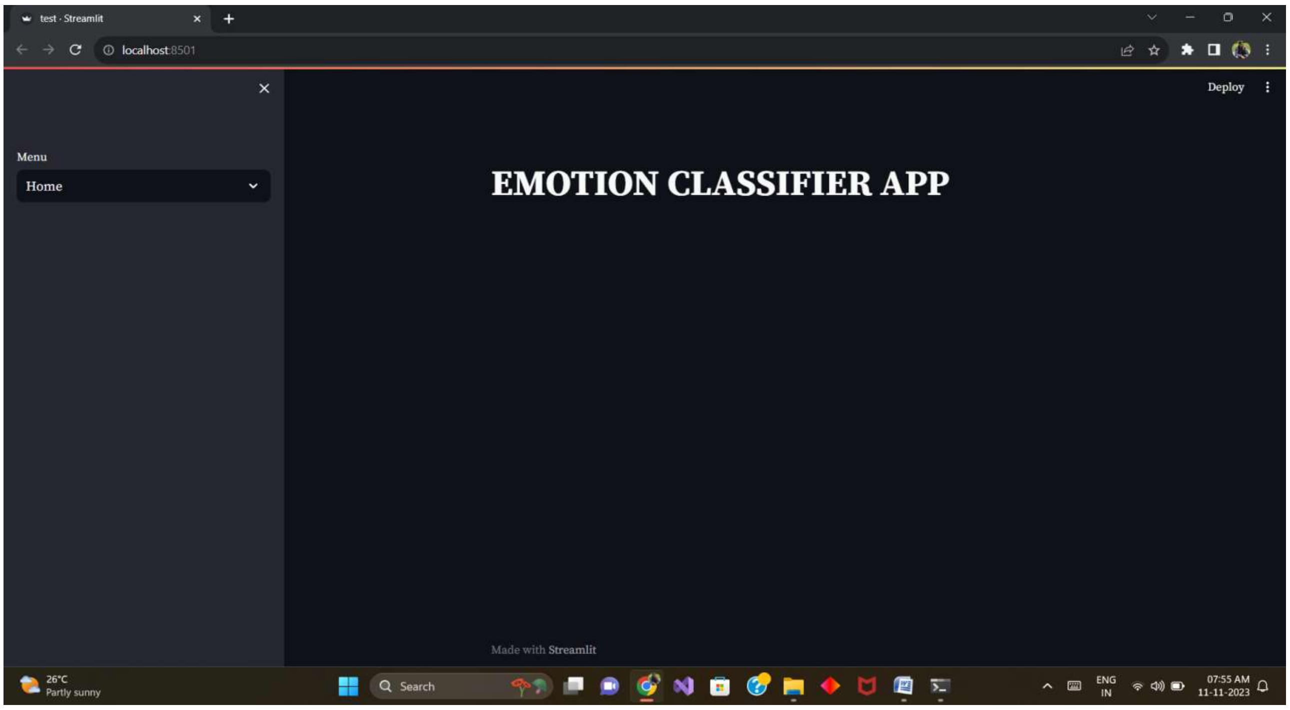
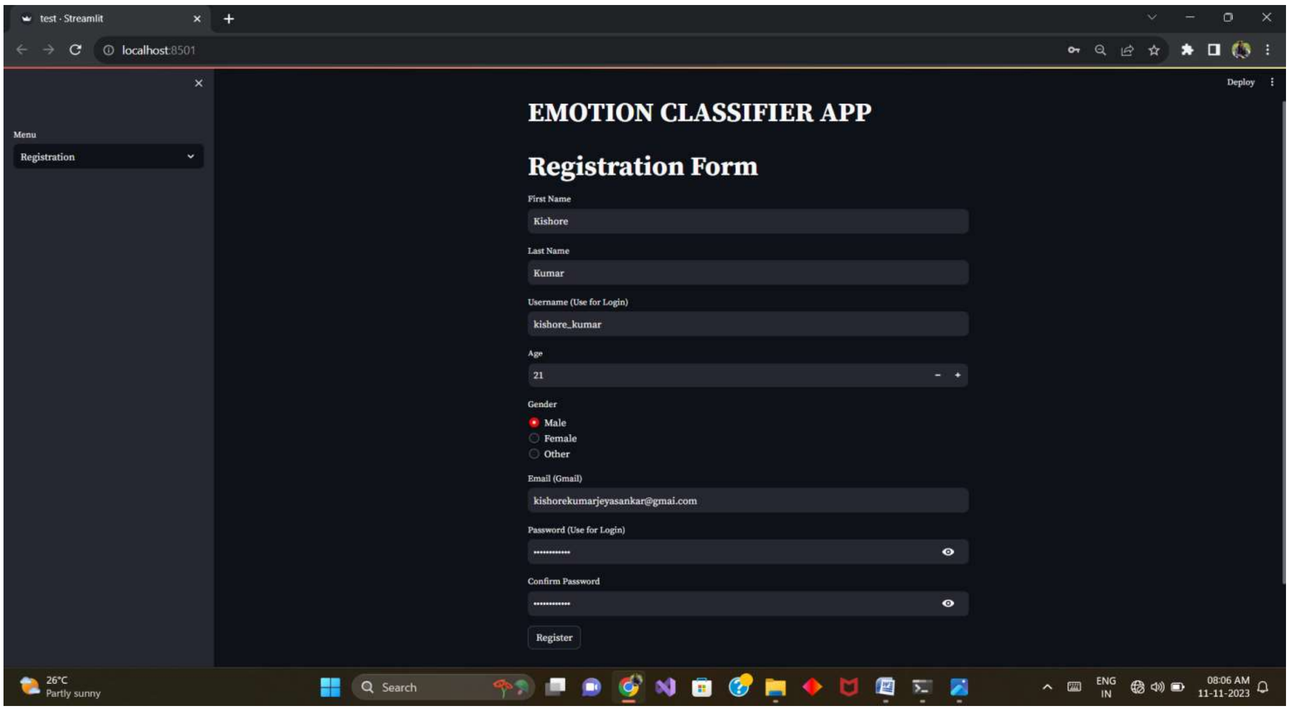


Figure A.1: HOME PAGE



EMOTION CLASSIFIER APP

Registration Form

First Name: Kishore

Last Name: Kumar

Username (Use for Login): kishore_kumar

Age: 21

Gender: [✓ Male] Female Other

Email (Gmail): kishorekumarjeyasankar@gmai.com

Password (Use for Login): ••••••••••••

Confirm Password: ••••••••••••

Register

Figure A.2: REGISTRATION PAGE

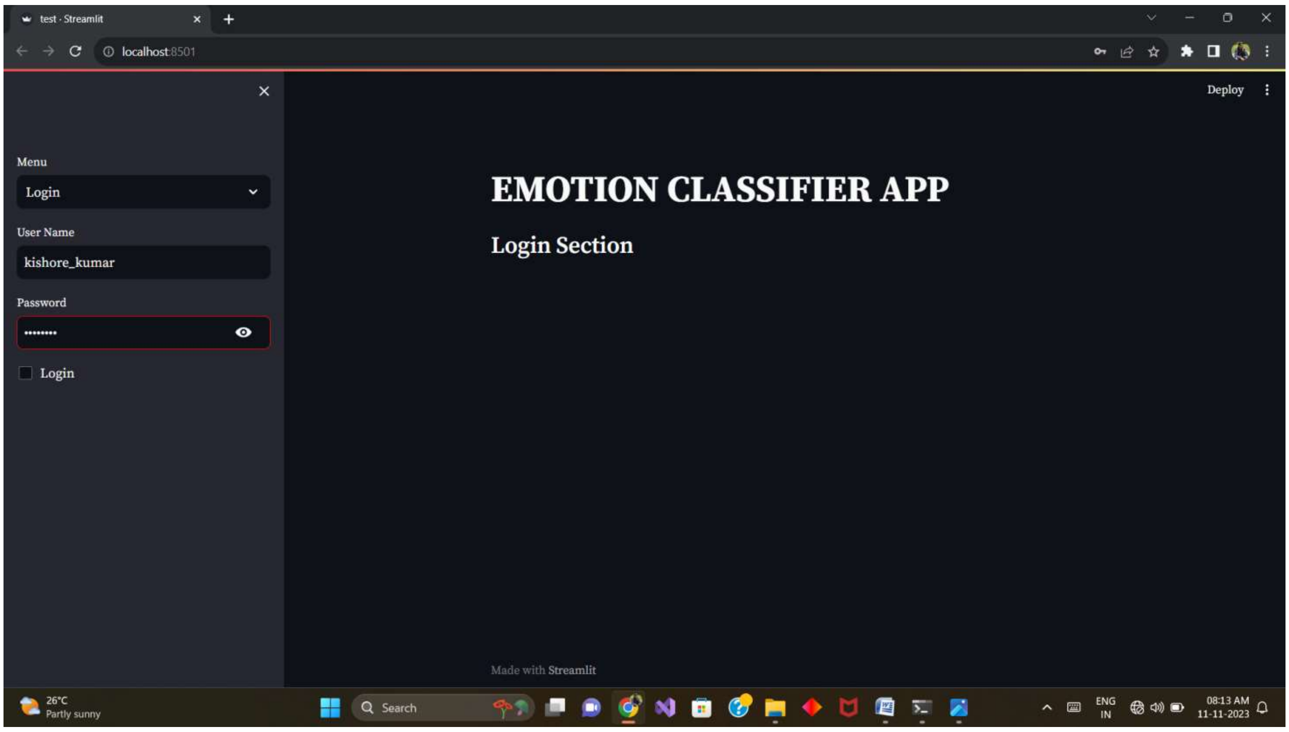
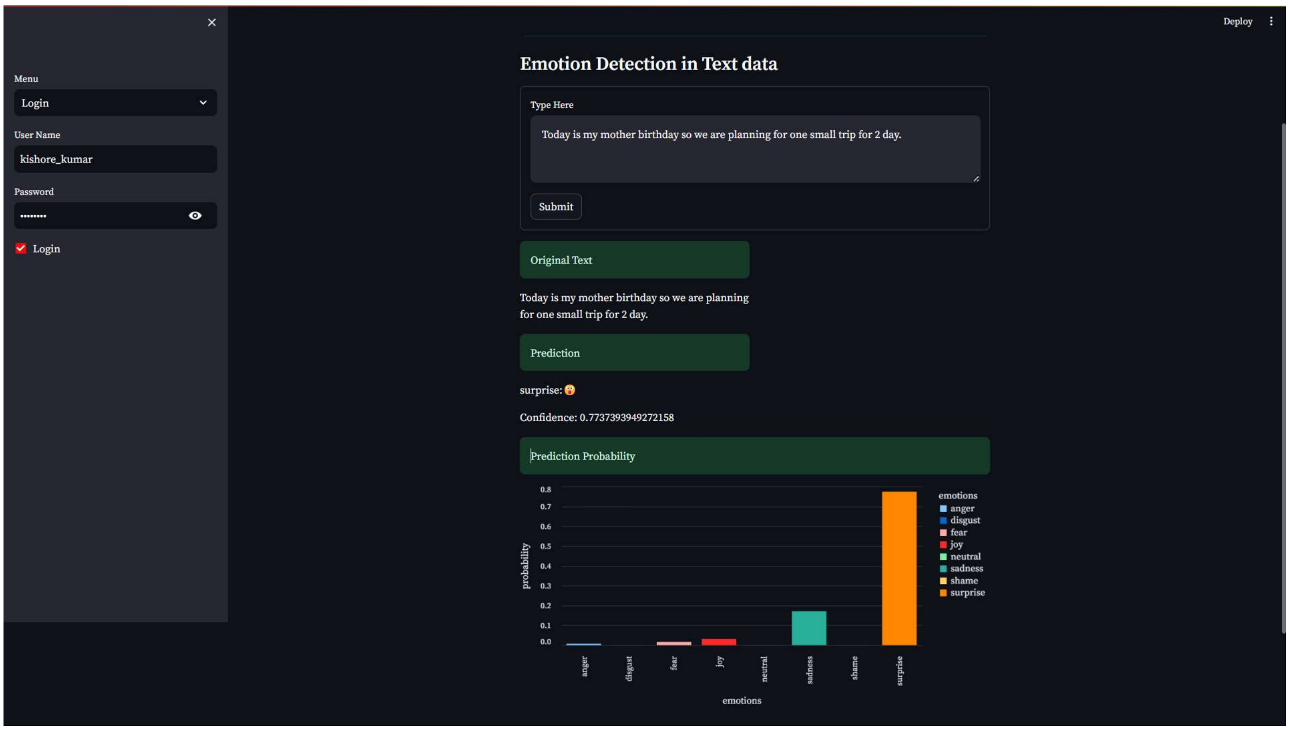


Figure B.1: LOGIN PAGE



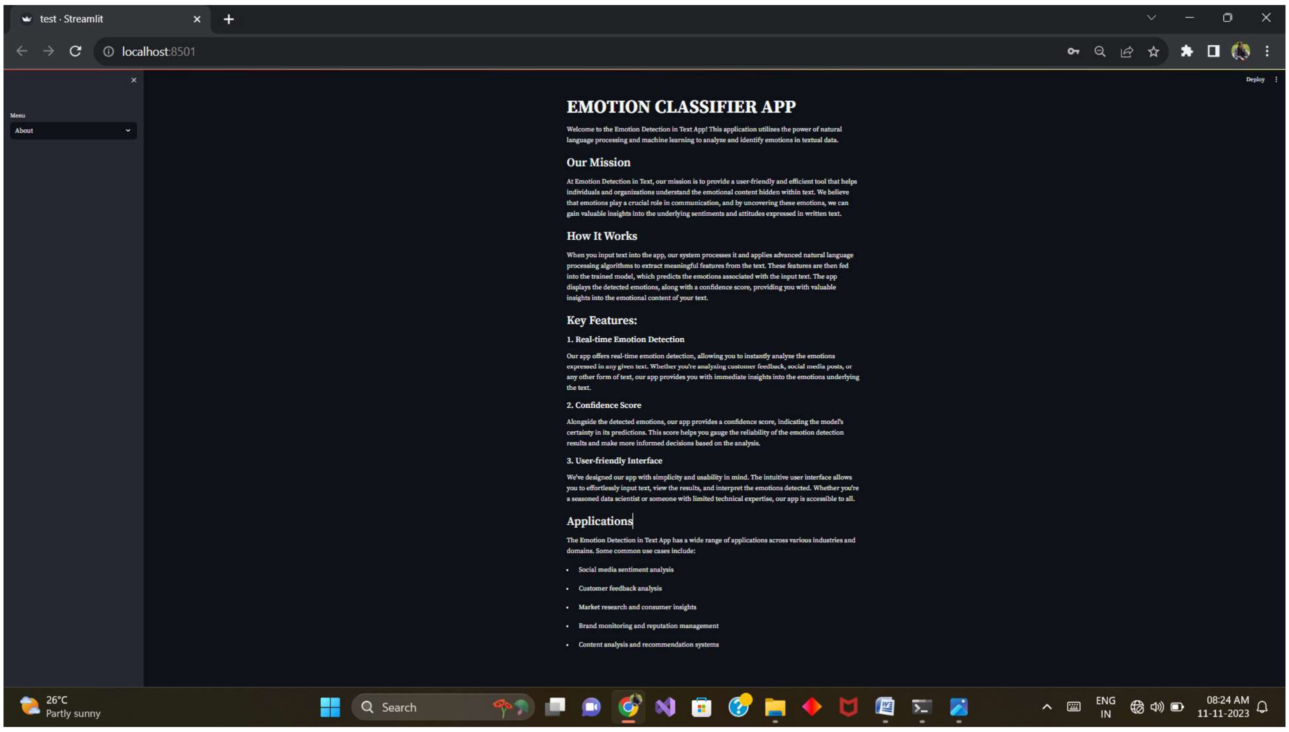
Today is my mother birthday so we are planning for one small trip for 2 day.

Today is my mother birthday so we are planning for one small trip for 2 day.

surprise:😲

Confidence: 0.7737393949272158

Figure B.2: EMOTION DETECTION



EMOTION CLASSIFIER APP

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.

Our Mission

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.

How It Works

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.

Key Features:

1. Real-time Emotion Detection

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.

2. Confidence Score

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.

3. User-friendly Interface

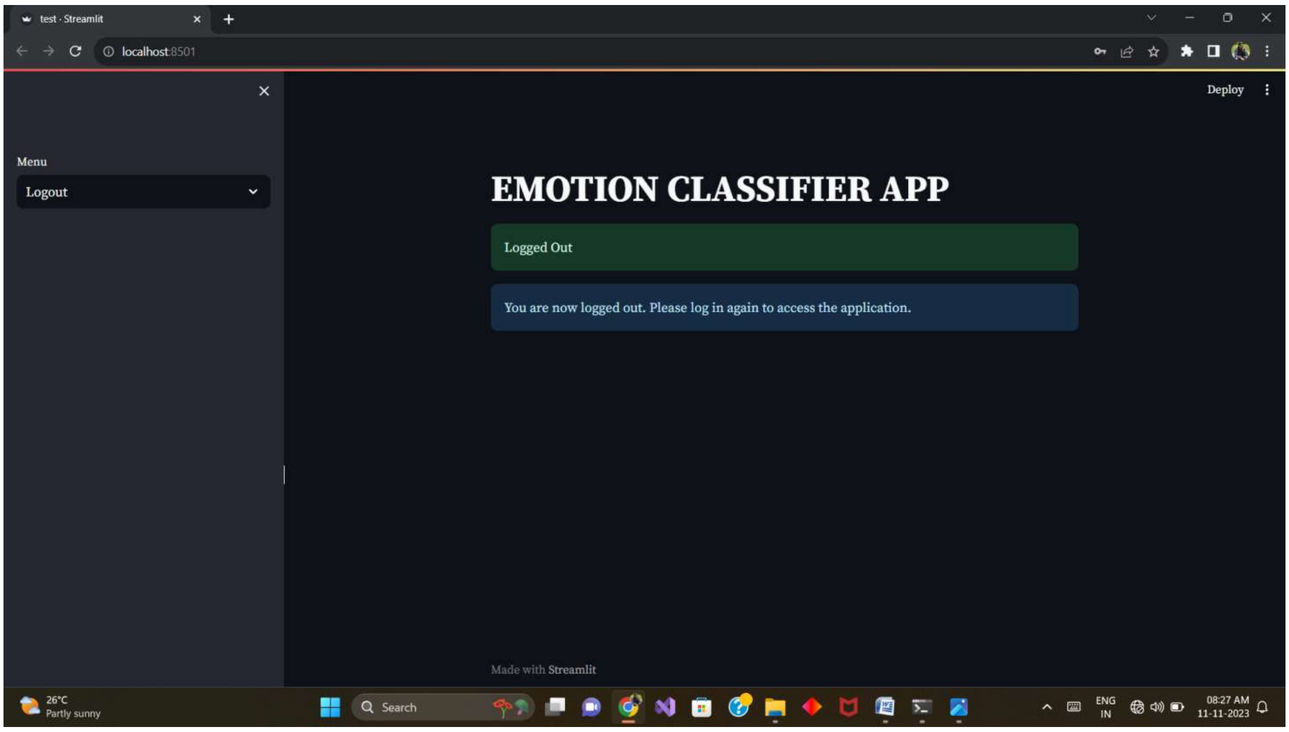
We've designed our app with simplicity and usability in mind. The intuitive user interface allows you to effortlessly input text, view the results, and interpret the emotions detected. Whether you're a seasoned data scientist or someone with limited technical expertise, our app is accessible to all.

Applications

The Emotion Detection in Text App has a wide range of applications across various industries and domains. Some common use cases include:

- Social media sentiment analysis
- Customer feedback analysis
- Market research and consumer insights
- Brand monitoring and reputation management
- Content analysis and recommendation systems

Figure C.1: ABOUT

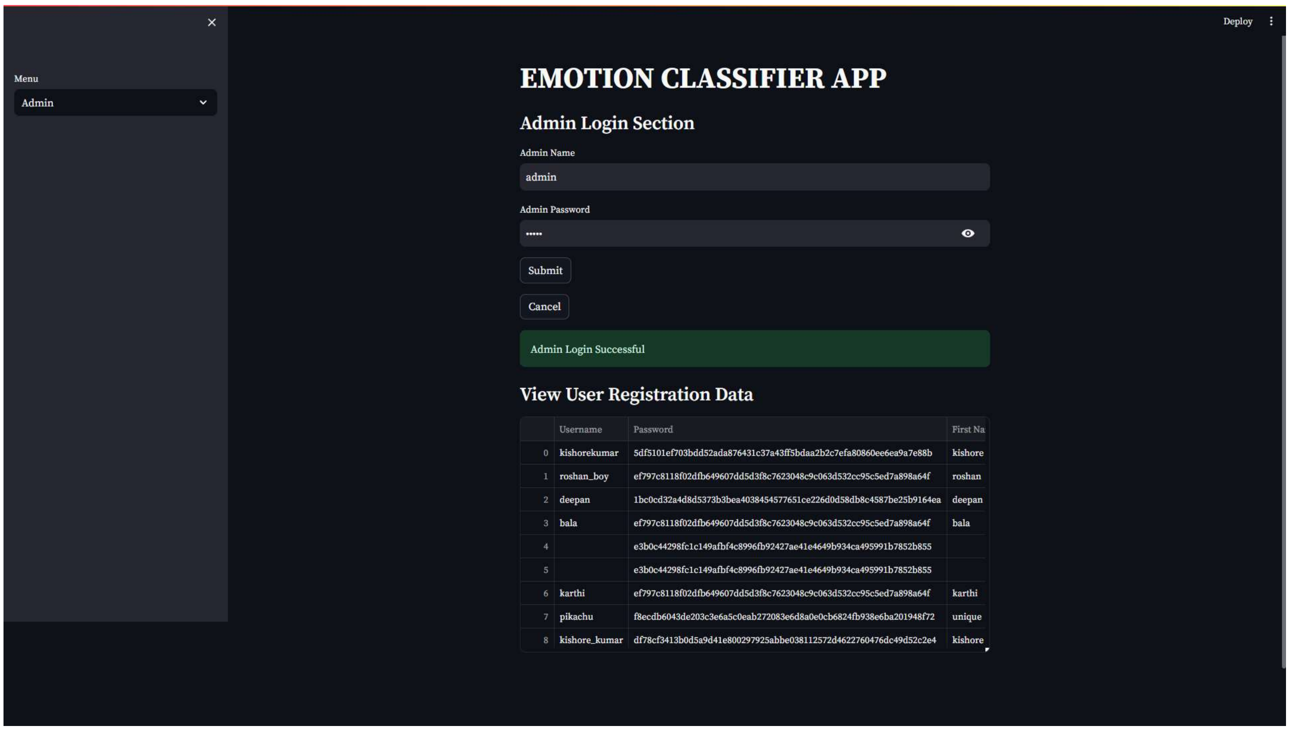


EMOTION CLASSIFIER APP

Logged Out

You are now logged out. Please log in again to access the application.

Figure C.2: LOGOUT



EMOTION CLASSIFIER APP

Admin Login Section

Admin Name: admin

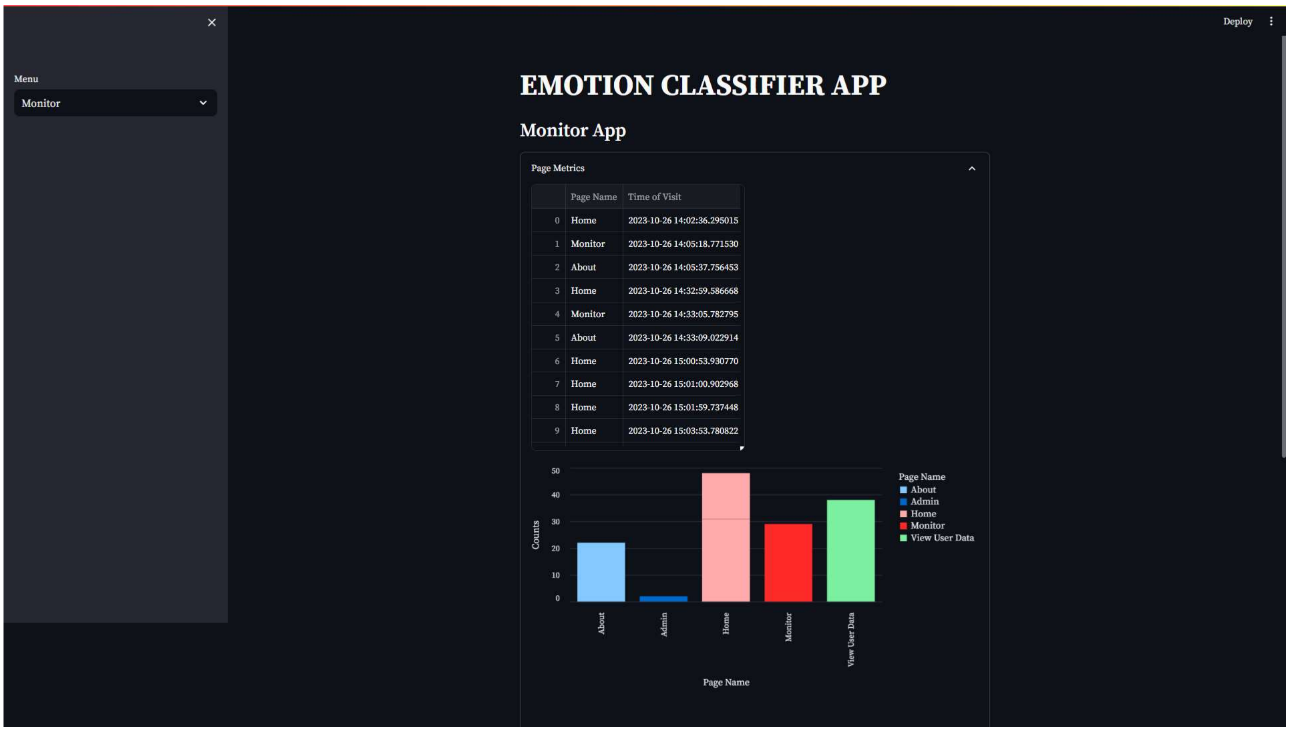
Admin Password: •••••

Admin Login Successful

View User Registration Data

	Username	Password	First Na
0	kishorekumar	5df5101ef703bdd52ada876431c37a43ff5bdaa2b2c7efa80860ee6ea9a7e88b	kishore
1	roshan_boy	ef797c8118f02dfb649607dd5d3f8c7623048c9c063d532cc95c5ed7a898a64f	roshan
2	deepan	1bc0cd32a4d8d5373b3bea4038454577651ce226d0d58db8c4587be25b9164ea	deepan
3	bala	ef797c8118f02dfb649607dd5d3f8c7623048c9c063d532cc95c5ed7a898a64f	bala
4		e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	
5		e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	
6	karthi	ef797c8118f02dfb649607dd5d3f8c7623048c9c063d532cc95c5ed7a898a64f	karthi
7	pikachu	f8ecdb6043de203c3e6a5c0eab272083e6d8a0e0cb6824fb938e6ba201948f72	unique
8	kishore_kumar	df78cf3413b0d5a9d41e800297925abbe038112572d4622760476dc49d52c2e4	kishore

Figure D.1: ADMIN



EMOTION CLASSIFIER APP

Monitor App

Page Metrics

	Page Name	Time of Visit
0	Home	2023-10-26 14:02:36.295015
1	Monitor	2023-10-26 14:05:18.771530
2	About	2023-10-26 14:05:37.756453
3	Home	2023-10-26 14:32:59.586668
4	Monitor	2023-10-26 14:33:05.782795
5	About	2023-10-26 14:33:09.022914
6	Home	2023-10-26 15:00:53.930770
7	Home	2023-10-26 15:01:00.902968
8	Home	2023-10-26 15:01:59.737448
9	Home	2023-10-26 15:03:53.780822

Figure D.2: MONITOR

CHAPTER 10
REFERENCES

BOOK REFERENCES

- "Natural Language Processing in Action" by Lane, Howard, and Hapke
- "Speech and Language Processing" by Jurafsky & Martin
- "Applied Natural Language Processing with Python" by Lane, Gauthier, and Howard
- "Foundations of Statistical Natural Language Processing" by Manning and Schütze
- "Emotion AI: The Intersection of Emotion and Artificial Intelligence" by Rana el Kaliouby
- "Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data" by Dipanjan Sarkar
- "Python Natural Language Processing" by Jalaj Thanaki
- "Mastering Natural Language Processing with Python" by Deepti Chopra, Nisheeth Joshi, and Iti Mathur

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.