SENTIMENT ANALYSIS CHATBOT
A PROJECT REPORT

Submitted by

DHARANIKA U **92132223035**
HAMSAVARTHINI S **92132223045**

MINI-PROJECT: SENTIMENT ANALYSIS CHATBOT

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University, Chennai)

DINDIGUL - 624622

OCTOBER 2024

PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY,

(Autonomous Institution Affiliated to Anna University, Chennai)

DINDIGUL – 624622

BONAFIDE CERTIFICATE

Certified that this idea report "**SENTIMENT ANALYSIS CHATBOT** " is the bonafide work of **"DHARANIKA U (92132223035), HAMSAVARTHINI S (92132223045)**" who carried out the idea work under my supervision .

SIGNATURE	SIGNATURE
Dr. A. VINCENT ANTONY KUMAR, M.E, Ph.D., **HEAD OF THE DEPARTMENT** PROFESSOR & HEAD DEPARTMENT OF IT PSNA COLLEGE OF ENGINEERING TECHNOLOGY, DINDIGUL -624622	**Dr. P. PRIYADHARSHINI M.E** ASSISTANT PROFESSOR DEPARTMENT OF IT PSNA COLLEGE OF ENGINEERING TECHNOLOGY, DINDIGUL -624622

Submitted for the idea on_____

ABSTRACT

This project report details the design and implementation of a sentiment analysis chatbot aimed at improving user interactions by accurately interpreting emotional nuances in text-based communications. In today's digital landscape, the ability to understand user sentiment is crucial for enhancing customer engagement, providing effective support, and fostering positive user experiences. This chatbot serves as a proactive tool for businesses to respond appropriately to user emotions.

The project is structured into several key phases. Initially, the purpose of the chatbot was defined, focusing on applications such as customer service, feedback collection, and social media interactions. Following this, a comprehensive technology stack was selected, incorporating advanced Natural Language Processing (NLP) frameworks and machine learning libraries, such as NLTK

A significant component of the project was the development of a sentiment analysis model. This involved gathering and preprocessing a diverse dataset with labeled sentiments, which included various text sources to ensure a wide-ranging understanding of emotional expressions. Utilizing machine learning algorithms, the model was trained to classify sentiments as positive, negative, or neutral, achieving a high degree of accuracy.

The integration of this model into the chatbot framework was pivotal. It enabled the chatbot to analyze user inputs in real-time, adapting conversation flows based on the detected sentiment. For instance, positive sentiments prompted engaging and encouraging responses, while negative sentiments initiated a more empathetic approach, offering solutions or assistance.

The deployment of the sentiment analysis chatbot has demonstrated its effectiveness in fostering meaningful interactions, significantly enhancing user satisfaction. This report encapsulates the methodologies employed, challenges faced during development, and the implications of sentiment analysis in conversational agents. Future recommendations include exploring advanced deep learning techniques and expanding the chatbot's capabilities to support multilingual sentiment analysis, further broadening its applicability across diverse user bases.

INTRODUCTION:

In an increasingly digital world, effective communication between businesses and users has become paramount. As customers engage with companies through various platforms—ranging from social media to direct messaging—understanding their emotions and sentiments can significantly influence satisfaction and loyalty. Traditional customer service approaches often fall short in addressing the nuanced emotional states of users, highlighting the need for more sophisticated interaction models. This is where sentiment analysis, combined with chatbot technology, plays a critical role.

Sentiment analysis, a subset of Natural Language Processing (NLP), involves the identification and classification of emotional tone in written text. By leveraging machine learning algorithms, businesses can gain insights into customer feelings, allowing for timely and appropriate responses. When integrated into chatbot systems, sentiment analysis enhances the ability to engage with users on a more personal level, transforming standard automated responses into meaningful conversation

PROBLEM STATEMENT:

As developers, we often seek to create applications that provide accurate information and connect with users on a deeper level. Traditional chatbots must improve at delivering empathetic and relevant responses, mainly when user emotions come into play. This project addresses the challenge of building a chatbot that understands the sentiments behind user messages and tailors its responses accordingly.

Let's say you have a friend to whom you can relate all that troubles you, and you suddenly talk to them, and they respond in a way that doesn't resonate with your emotions. Of course, you'd feel some amount of disappointment. Combining sentiment analysis and rule-based response generation, we aim to enhance the user experience and create a more engaging conversational environment.

CHALLENGES:

The development of a sentiment analysis chatbot presents several challenges that need to be addressed to ensure its effectiveness and reliability. These challenges encompass technical, linguistic, and operational aspects:

1. Complexity of Natural Language: Human language is inherently complex, characterized by nuances such as sarcasm, idioms, and context-dependent meanings. Accurately capturing these subtleties poses a significant challenge for sentiment analysis models, which may struggle to classify sentiments correctly.
2. Variability in User Expressions: Users express emotions in diverse ways, influenced by factors like

culture, age, and personal experiences. Creating a model that can generalize across various expressions while maintaining high accuracy requires extensive training data and continuous refinement.

3. Data Quality and Quantity: The performance of the sentiment analysis model relies heavily on the quality and volume of training data. Acquiring large, well-annotated datasets can be resource-intensive, and biased or insufficient data may lead to inaccurate sentiment classification.
4. Integration with Existing Systems: Seamlessly integrating the sentiment analysis model into an existing chatbot framework can be technically challenging. Ensuring that the model operates in real-time and communicates effectively with other components requires careful design and implementation.

PROPOSED MODEL:

The proposed sentiment analysis chatbot model integrates advanced Natural Language Processing (NLP) techniques with machine learning algorithms to effectively analyze user sentiments and deliver contextual responses. The process begins with **data collection and preprocessing**, where diverse text data is gathered from sources like social media and customer reviews, annotated for sentiments, and cleaned through tokenization and normalization. For the **sentiment analysis model**, a hybrid approach is used, combining traditional machine learning algorithms with deep learning techniques, particularly leveraging pre-trained transformer models like BERT to capture contextual relationships in text.

Integration into a **chatbot framework** involves **Natural Language Understanding (NLU)** to interpret user inputs and **dialogue management** to maintain coherent conversations. The chatbot generates **sentiment-aware responses**, tailoring interactions based on the identified emotional tone. To ensure continuous improvement, a **user feedback mechanism** allows users to rate responses, providing valuable data for ongoing model refinement. Finally, **evaluation metrics** such as accuracy, precision, recall, and user satisfaction surveys are employed to assess performance. Overall, this model aims to create a robust sentiment analysis chatbot that enhances customer engagement and fosters stronger relationships between businesses and their clients.

SOURCE CODE:

Backend code:(python)

```
from flask import Flask, request, jsonify, render_template
app = Flask(__name__)

greetings = ['hi', 'hello', 'hey', 'howdy']
farewells = ['bye', 'goodbye', 'see you', 'take care']
positive_words = ['good', 'great', 'awesome', 'happy', 'love', 'fantastic', 'excellent', 'amazing', 'wonderful',
```

```
'joyful']
negative_words = ['bad', 'sad', 'hate', 'angry', 'terrible', 'horrible', 'awful', 'disappointing', 'frustrating', 'unhappy']

def analyze_sentiment(user_input):
    score = 0
    user_words = user_input.lower().split()

    for word in user_words:
        if word in positive_words:
            score += 1
        elif word in negative_words:
            score -= 1

    return score

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/get_response', methods=['POST'])
def get_response():
    user_input = request.json['user_input']
    bot_response = ""

    # Check for greetings
    if any(greet in user_input.lower() for greet in greetings):
        bot_response = "Hello! How can I assist you today?"
    # Check for farewells
    elif any(farewell in user_input.lower() for farewell in farewells):
        bot_response = "Goodbye! Have a great day!"
    # Simple sentiment analysis
    else:
        score = analyze_sentiment(user_input)

        if score > 2:
            bot_response = "That's wonderful to hear! Keep spreading the positivity!"
        elif score > 0:
```

```
            bot_response = "I'm glad to hear you're feeling positive!"
        elif score < -2:
            bot_response = "I'm really sorry to hear you're feeling that way. It's okay to feel down sometimes."
        elif score < 0:
            bot_response = "I'm sorry to hear you're feeling negative. If you want to talk about it, I'm here!"
        else:
            bot_response = "It seems like you have mixed feelings. Would you like to share more?"

    return jsonify({'bot_response': bot_response})

if __name__ == '__main__':
    app.run(debug=True)
```

Frontend Code:(HTML,Css)

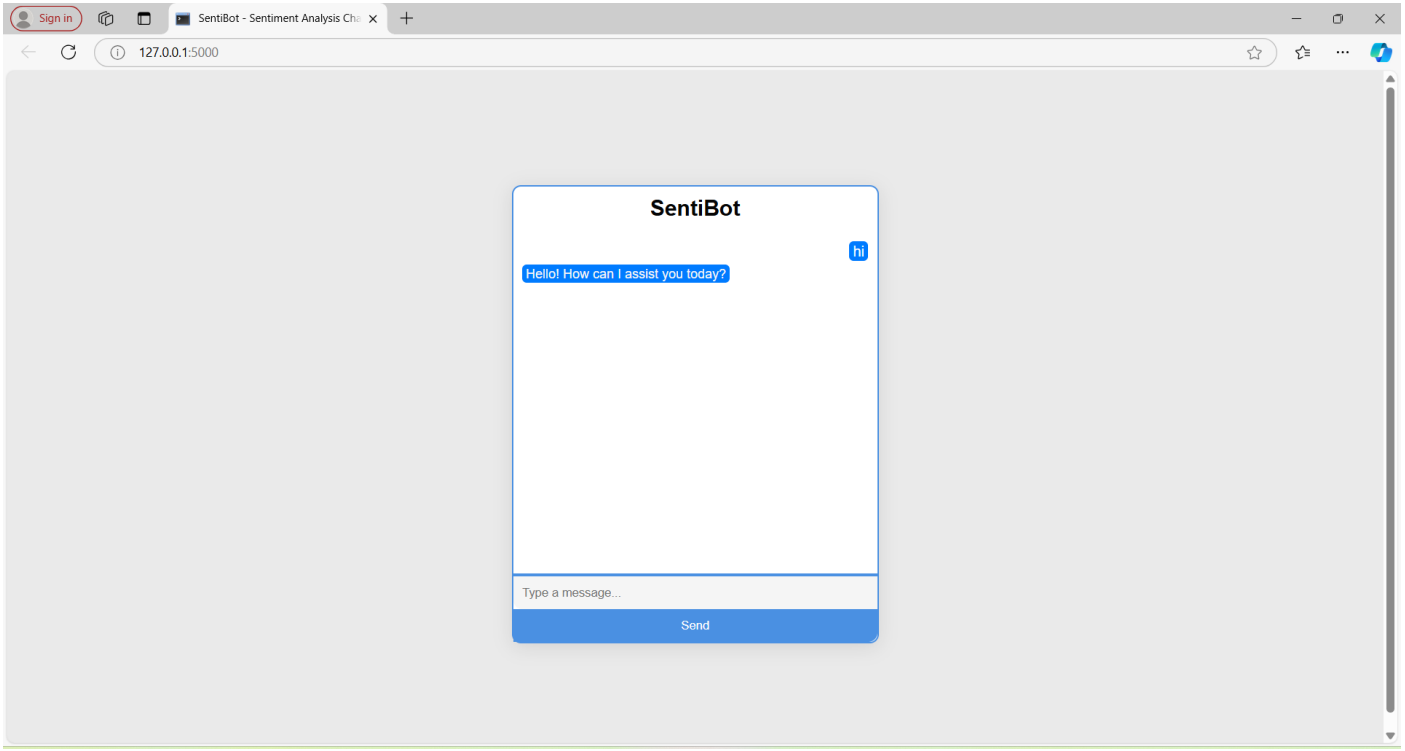
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="new1.css">
    <title>SentiBot - Sentiment Analysis Chatbot</title>
</head>
<style>
    body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #eaeaea;
}

.chat-container {
    border: 2px solid #4a90e2;
    border-radius: 10px;
    width: 400px;
    height: 500px;
```

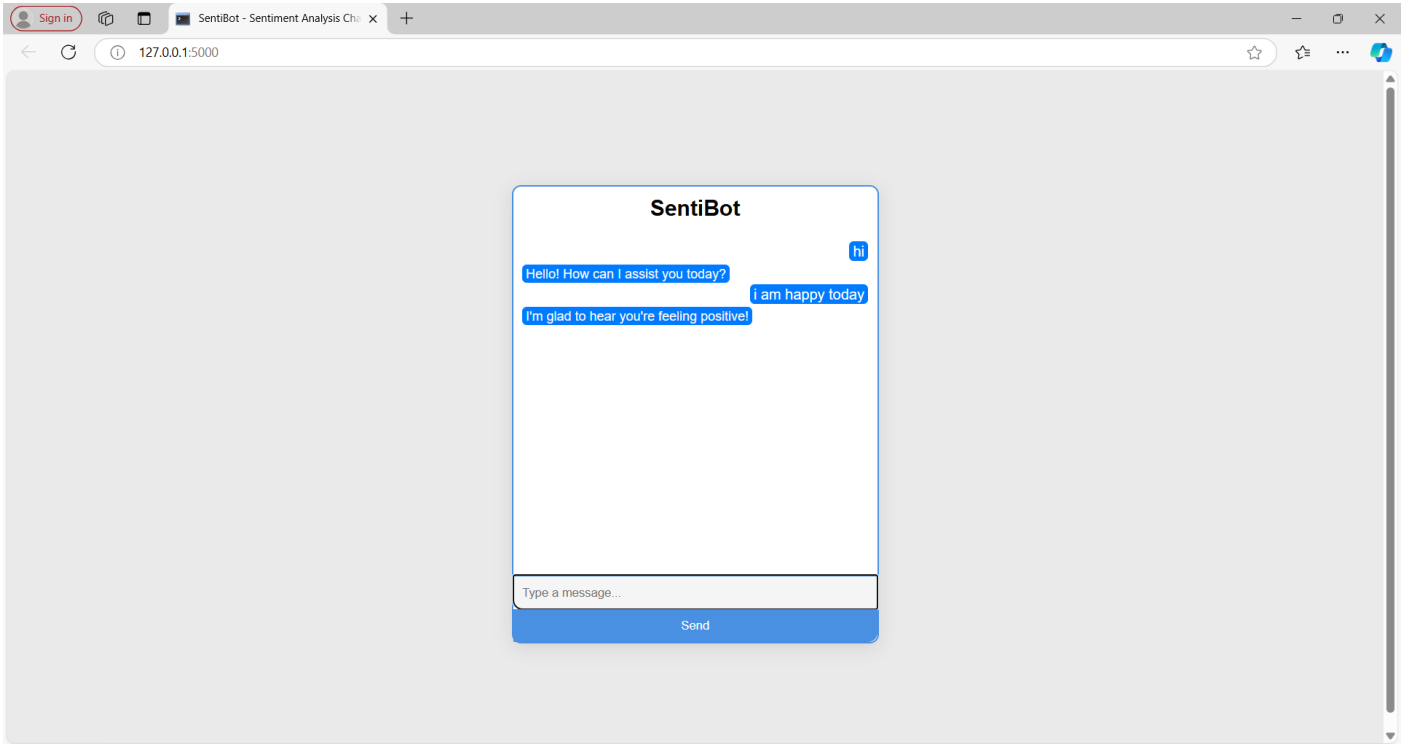
```
    display: flex;
    flex-direction: column;
    background-color: #ffffff;
    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);
}

.chat-title {
    text-align: center;
    color: #050505;
    margin: 10px 0;
}

.chat-box {
    flex: 1;
    padding: 10px;
    overflow-y: auto;
    border-bottom: 2px solid #4a90e2;
}

input {
    padding: 10px;
    border: none;
    border-top: 2px solid #4a90e2;
    flex: 0;
    border-radius: 0 0 0 10px;
    background-color: #f5f5f5; /* Background color for input */
}

button {
    padding: 10px;
    border: none;
    background-color: #4a90e2;
    color: white;
    cursor: pointer;
    border-radius: 0 0 10px 0;
}
```

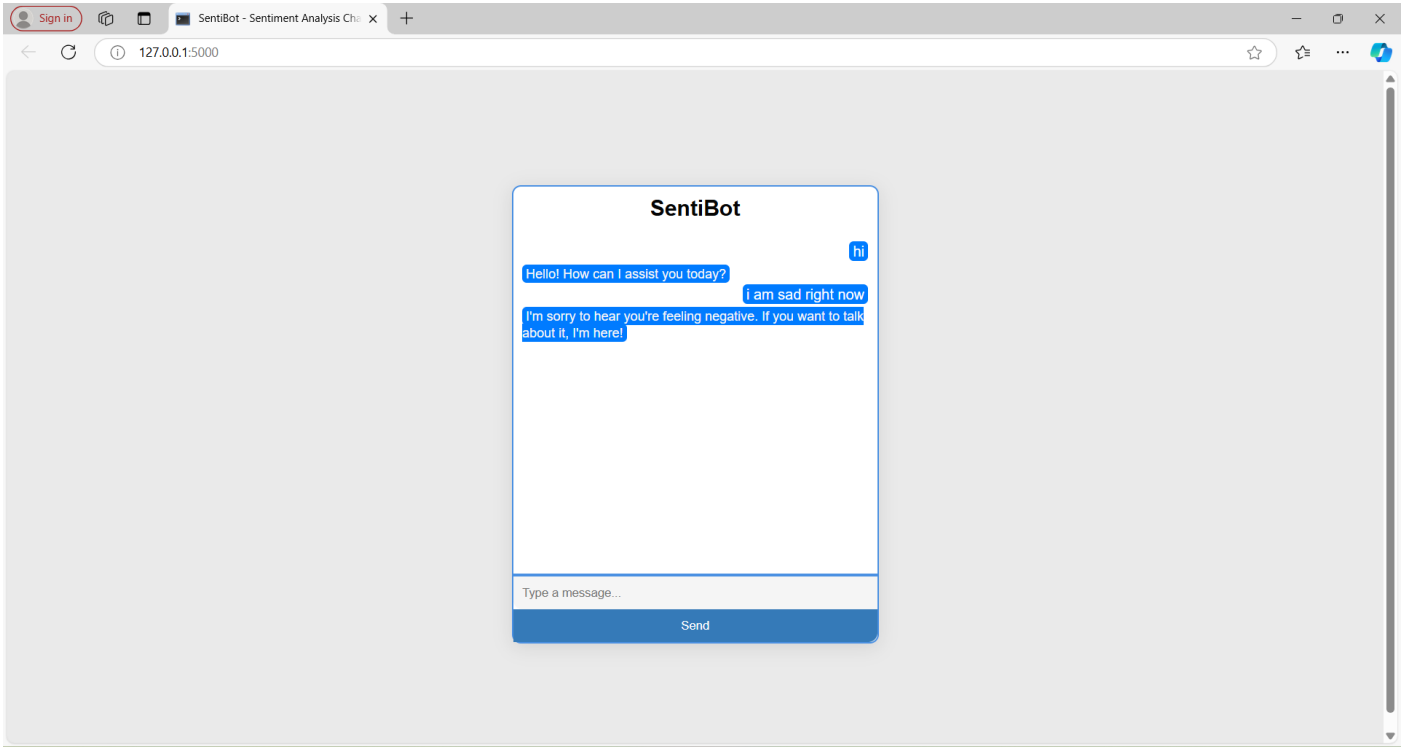
```
button:hover {
    background-color: #357ab8;
}



.user-message {
    text-align: right;
    margin: 5px 0;
}

.user-text {
    color: rgb(255, 255, 255); /* Text color for user messages */
    background-color: #007bff; /* Background color for user messages */
    padding: 2px 4px; /* Padding to adjust the height */
    border-radius: 5px; /* Rounded corners for user messages */
}

.bot-message {
    text-align: left;
    margin: 5px 0;
}

.bot-text {
    color: rgb(248, 246, 246); /* Text color for bot messages */
    background-color: #007bff; /* Background color for bot messages */
    padding: 2px 4px; /* Padding to adjust the height */
    border-radius: 5px; /* Rounded corners for bot messages */
    font-size: 0.9em; /* Smaller font size for bot messages */
}

</style>
<body>
    <div class="chat-container">
        <h2 class="chat-title">SentiBot</h2>
        <div class="chat-box" id="chat-box"></div>
        <input type="text" id="user-input" placeholder="Type a message..." />
        <button id="send-btn">Send</button>
```

```
    </div>

</body>
<script>
    document.getElementById('send-btn').addEventListener('click', function() {
        const userInput = document.getElementById('user-input').value;
        if (userInput.trim() === "") return;

        // Send user input to Flask backend
        fetch('/get_response', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify({ user_input: userInput })
        })
        .then(response => response.json())
        .then(data => {
            // Update chat box with bot response
            const chatBox = document.getElementById('chat-box');
            chatBox.innerHTML += `<div class="user-message"><span class="usertext">
            ${userInput}</span></div>`;
            chatBox.innerHTML += `<div class="bot-message"><span class="bottext">
            ${data.bot_response}</span></div>`;
            document.getElementById('user-input').value = ''; // Clear input field
        })
        .catch(error => console.error('Error:', error));
    });
</script>

</html>
```

OUTPUT:


Sign in
SentiBot - Sentiment Analysis Cha
127.0.0.1:5000
SentiBot
hi
Hello! How can I assist you today?
Type a message...
Send


Sign in
SentiBot - Sentiment Analysis Cha
127.0.0.1:5000
SentiBot
hi
Hello! How can I assist you today?
i am happy today
I'm glad to hear you're feeling positive!
Type a message...
Send



CONCLUSION:

In summary, the sentiment analysis chatbot significantly enhances user interactions by leveraging advanced Natural Language Processing and machine learning techniques to accurately interpret and respond to user sentiment. This project showcases the transformative potential of sentiment analysis in conversational agents, paving the way for more intelligent customer service solutions. Future enhancements could include multilingual support and deeper contextual understanding, further enriching user engagement and driving business.

SENTIMENT ANALYSIS CHATBOT

Mini-Project Presentation

by
DHARANIKA U (92132223035)
HAMSAVARTHINI S (92132223045)

Abstract :

- A chatbot or conversational agent is a software that can interact or “chat” with a human user using a natural language, like English, for instance.
- Since the first chatbot developed, many have been created but most of their problems still persist, like providing the right answer to the user and user acceptance itself.
- Considering such facts, in this work, we present a chatbot-building framework that considers the use of sentiment analysis and tree timelines to provide a better chatbot answer.
- For instance, as presented in our experiments, the user can be addressed to a human attendant when its sentiment is very negative, as an alternative answer, whenever the user sentiment is less negative.

Introduction :

- Sentiment analysis is a popular task in natural language processing.
- The goal of sentiment analysis is to classify the text based on the mood or mentality expressed in the text, which can be positive negative, or neutral.
- The goal that Sentiment mining tries to gain is to be analysis people's opinions in a way that can help businesses expand. It focuses not only on polarity (positive, negative & neutral) but also on emotions (happy, sad, angry, etc.).
- It uses various Natural Language Processing algorithms such as Rule-based algorithm

Problem Statement :

- As developers, we often seek to create applications that provide accurate information and connect with users on a deeper level.
- Traditional chatbots must improve at delivering empathetic and relevant responses, mainly when user emotions come into play.
- This project addresses the challenge of building a chatbot that understands the sentiments behind user messages and tailors its responses accordingly.
- Combining sentiment analysis and rule-based response generation, we aim to enhance the user experience and create a more engaging conversational environment.

Challenges :

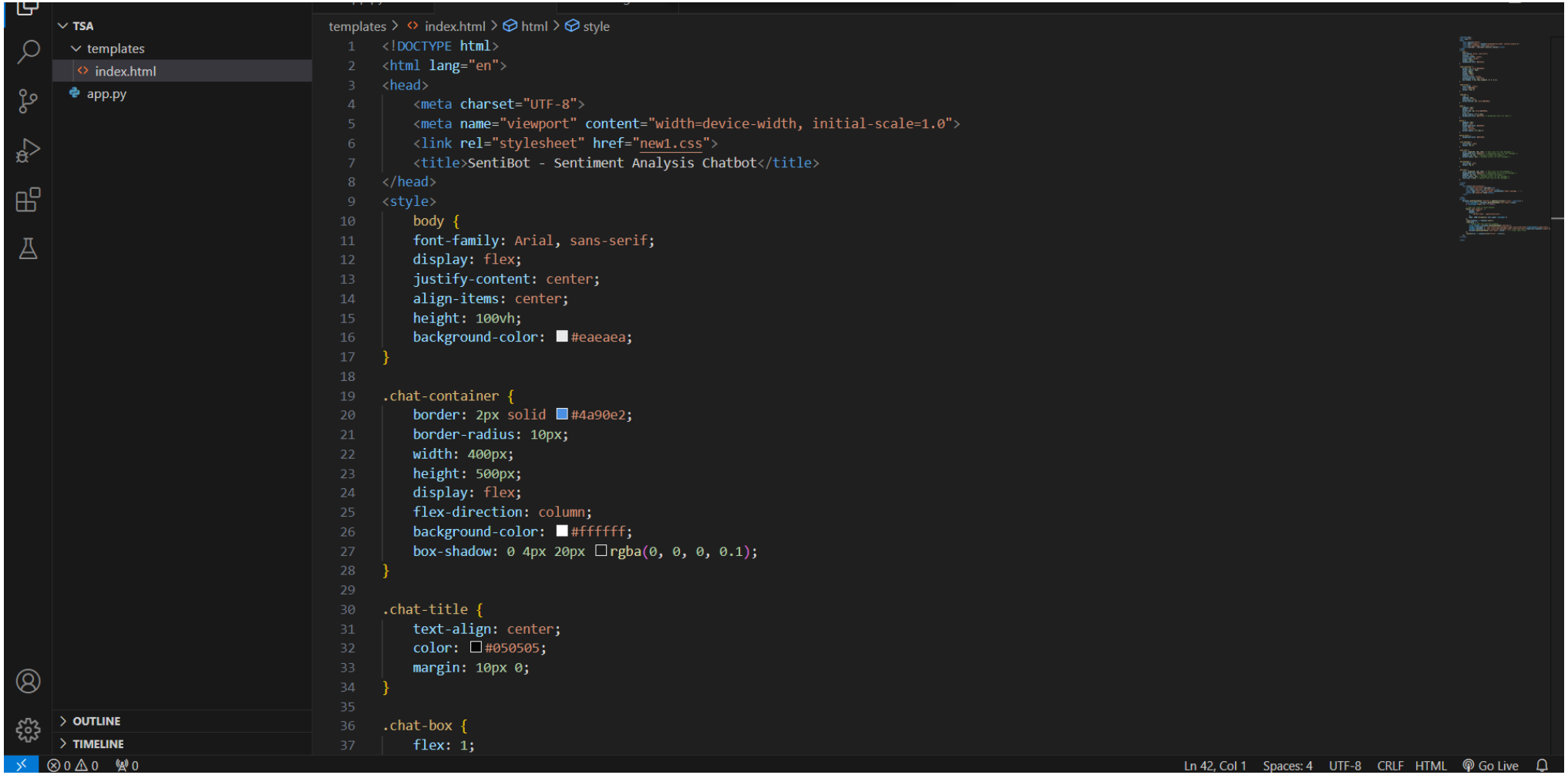
- Sentiment analysis include interpreting the meaning of words in different contexts
- **Data Quality:** Some user input had spelling errors or informal language, complicating sentiment classification.
- **Sarcasm and Contextual Understanding:** The model struggled to detect sarcasm and understand complex contexts.
- **Real-time Processing:** Ensuring low-latency response times when deployed at scale.
- **Imbalanced Data:** Neutral sentiment was underrepresented, leading to occasional misclassifications.

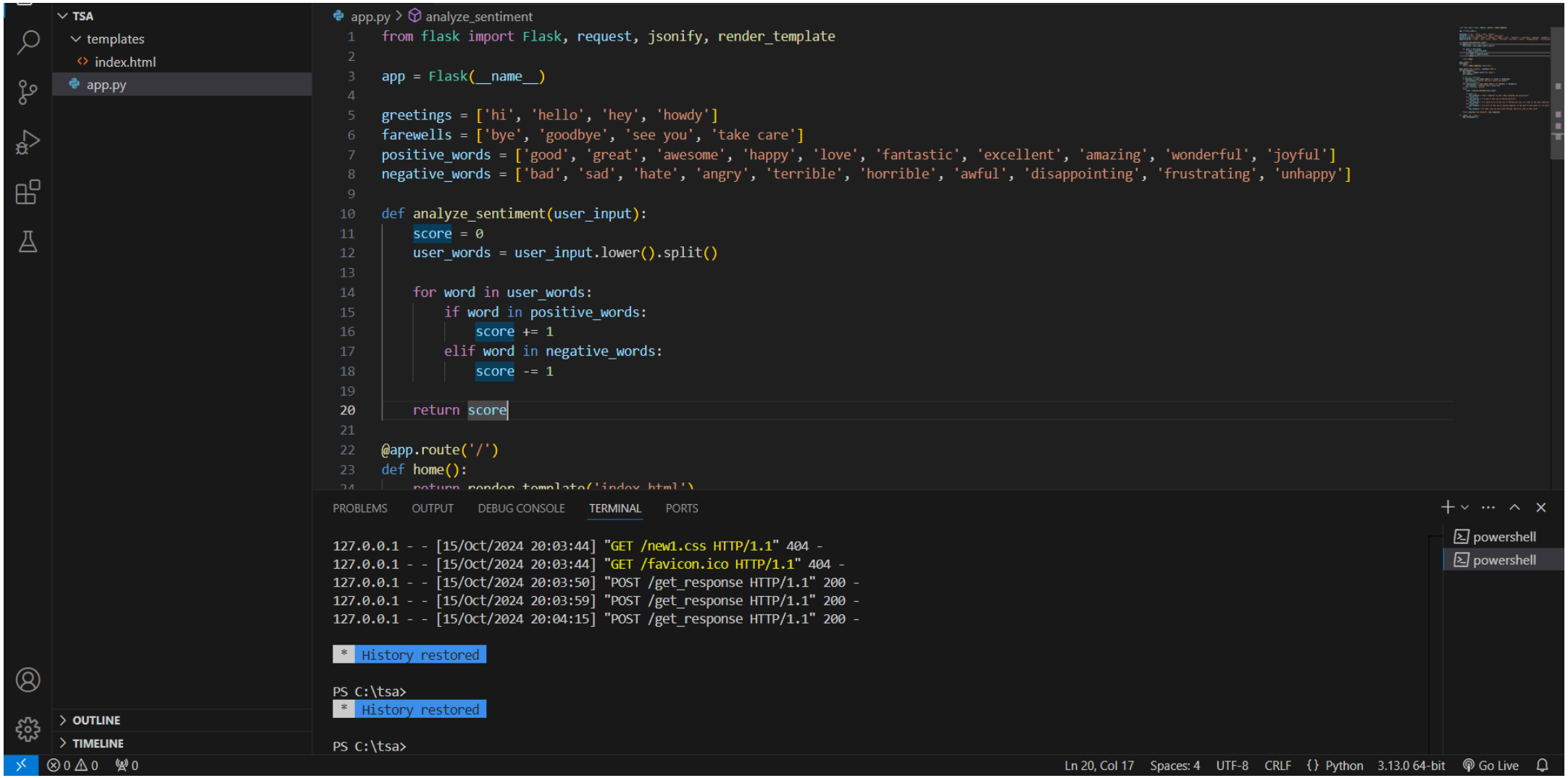
Proposed Model :

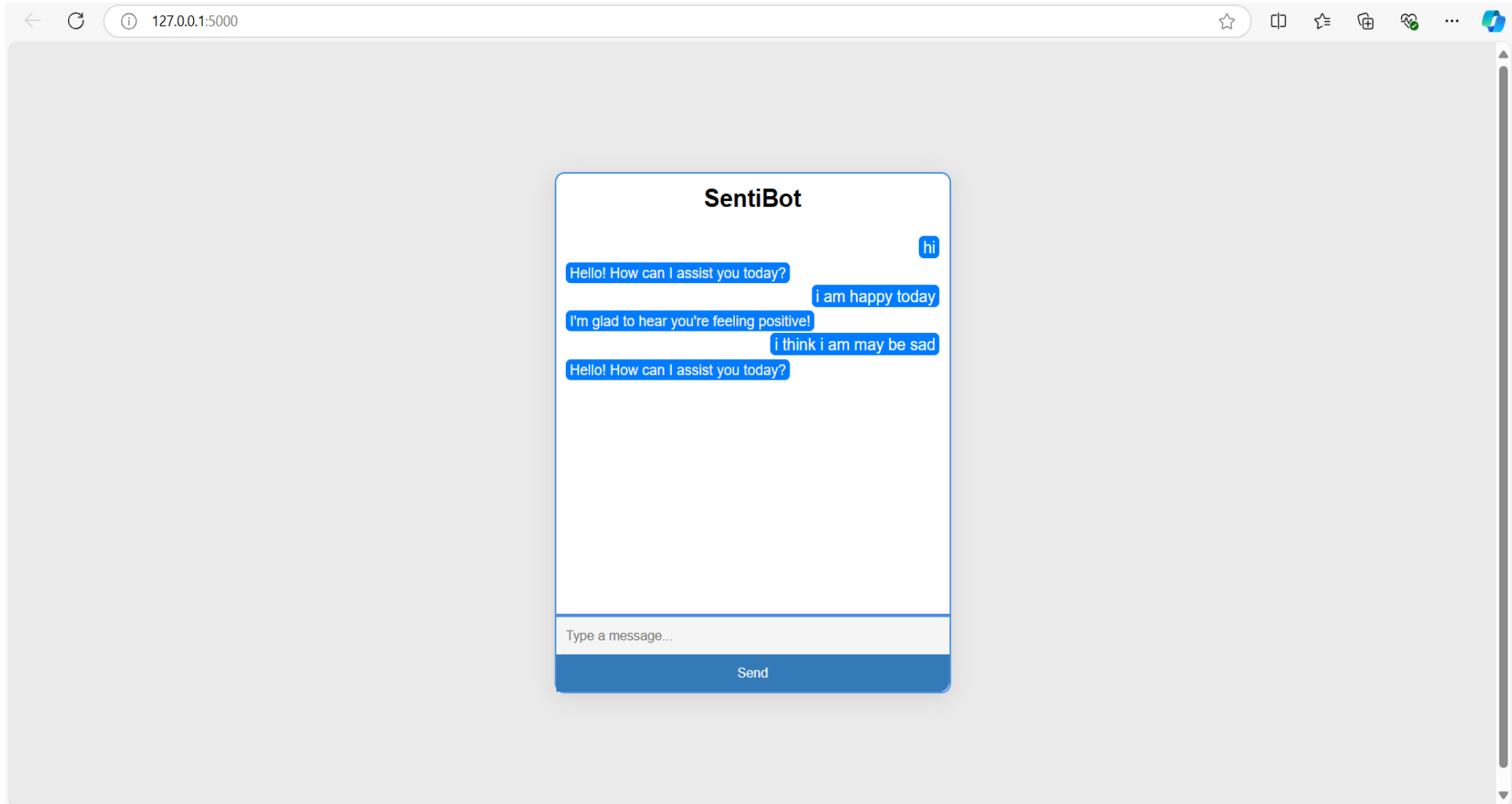
1. **Created a Chatbot**:
   - Developed a chatbot named SentiBot that can chat with users.
2. **Sentiment Analysis**:
   - The chatbot analyzes the feelings expressed in user messages.
   - It can identify positive and negative sentiments based on specific words.
3. **User Interaction**:
   - Users can type messages and receive replies in real time.
   - The chatbot greets users and says goodbye when they leave.

4. **Friendly Interface**:
   - Built a simple and attractive chat interface using HTML and CSS.
   - Users can easily type their messages and see the chatbot's responses.
5. **Flask Backend**:
   - Used Flask, a web framework, to handle user messages and send responses.
   - The chatbot runs on a web server, allowing users to interact with it via a web browser.
6. **Real-Time Responses**:
   - Implemented JavaScript to send messages to the server and update the chat without reloading the page.

Implementation / Code :




TSA
templates
index.html
app.py
app.py > analyze_sentiment
from flask import Flask, request, jsonify, render_template
app = Flask(__name__)
greetings = ['hi', 'hello', 'hey', 'howdy']
farewells = ['bye', 'goodbye', 'see you', 'take care']
positive_words = ['good', 'great', 'awesome', 'happy', 'love', 'fantastic', 'excellent', 'amazing', 'wonderful', 'joyful']
negative_words = ['bad', 'sad', 'hate', 'angry', 'terrible', 'horrible', 'awful', 'disappointing', 'frustrating', 'unhappy']
def analyze_sentiment(user_input):
score = 0
user_words = user_input.lower().split()
for word in user_words:
if word in positive_words:
score += 1
elif word in negative_words:
score -= 1
return score
@app.route('/')
def home():
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
127.0.0.1 - - [15/Oct/2024 20:03:44] "GET /new1.css HTTP/1.1" 404 -
127.0.0.1 - - [15/Oct/2024 20:03:44] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [15/Oct/2024 20:03:50] "POST /get_response HTTP/1.1" 200 -
127.0.0.1 - - [15/Oct/2024 20:03:59] "POST /get_response HTTP/1.1" 200 -
127.0.0.1 - - [15/Oct/2024 20:04:15] "POST /get_response HTTP/1.1" 200 -
* History restored
PS C:\tsa>
* History restored
PS C:\tsa>
powershell
powershell
OUTLINE
TIMELINE
Ln 20, Col 17
Spaces: 4
UTF-8
CRLF
Python
3.13.0 64-bit
Go Live


127.0.0.1:5000
SentiBot
hi
Hello! How can I assist you today?
i am happy today
I'm glad to hear you're feeling positive!
i think i am may be sad
Hello! How can I assist you today?
Type a message...
Send

Conclusion :

- Through sentiment analysis, you have an effective way that helps you to understand not only how your customers communicate but also the emotions behind what they say.
- It helps you find effective ways to improve the quality of service you deliver and reduce churn in the process.

THANK YOU