

SUMMER INTERSHIP PROJECT REPORT
ON
“Sentiment Analysis with Chatbot”
Submitted in partial fulfilment of the requirements for
the award of the degree of
Bachelor of Technology
In
Artificial Intelligence & Machine Learning
By

SIDDHARTHA
A501132522011

Under the guidance of
Dr. Sakshi Kathuria
Assistant Professor



Department of Computer Science and Engineering
Amity School of Engineering and Technology
Amity University Haryana
Gurgaon, India
August, 2025



Department of Computer Science and Engineering

Amity School of Engineering and Technology

Declaration

I, **SIDDHARTHA, A501132522011**, student of Bachelor of Technology (or Master of Computer Applications) in Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Haryana, hereby declare that I am fully responsible for the information and results provided in this project report titled **“Sentiment Analysis with Chatbot”** submitted Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Haryana, Gurgaon for the partial fulfilment of the requirement for the award of the degree of ***Bachelor of Technology in Artificial Intelligence & Machine Learning***. I have taken care in all respects to honour the intellectual property rights and have acknowledged the contributions of others for using them. I further declare that in case of any violation of intellectual property rights or copyrights, I as a candidate will be fully responsible for the same. My supervisor, Head of department and the Institute should not be held for full or partial violation of copyrights if found at any stage of my degree.

SIDDHARTHA

A501132522011



Department of Computer Science and Engineering

Amity School of Engineering and Technology

Certificate

This is to certify that the work in the project report entitled “**Sentiment Analysis with Chatbot**” by **SIDDHARTHA,A501132522011** is a bonafide record of project work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Artificial Intelligence & Machine Learning** in the **Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Haryana, Gurgaon**. Neither this project nor any part of it has been submitted for any degree or academic award elsewhere.

Date: 25th August 2025

Dr. Sakshi Kathuria

Assistant Professor

Computer Science & Engineering ASET, Amity University, Haryana

Head

Dr. Shalini Bhaskar Bajaj

Department of Computer Science & Engineering ,Amity School of Engineering and Technology Amity University Haryana

CERTIFICATE



ACKNOWLEDGEMENT

I wish to express my deepest gratitude for the successful completion of this *Sentiment Analysis with Chatbot project*, which was made possible through the unwavering support, encouragement, and guidance of several remarkable individuals. First and foremost, I extend my heartfelt thanks to **Dr. Sakshi Kathuria**, Assistant Professor at ASET, Amity University Haryana. Her expertise, valuable insights, and mentorship have been invaluable, providing me with the confidence and direction needed to tackle the challenges of this project.

I am also profoundly grateful to my family for their constant encouragement and moral support. Their belief in my abilities and patience during the demanding phases of development were instrumental in achieving this goal. Additionally, I wish to acknowledge the contributions of my friends and peers whose collaboration, constructive feedback, and enthusiasm greatly enhanced the refinement and effectiveness of this project.

In conclusion, I sincerely thank everyone who supported and contributed to the completion of this project. Your guidance, encouragement, and motivation have been a continuous source of strength and inspiration, and I am deeply appreciative of the collective effort that has led to the successful realization of this project.

Siddhartha

A501132522011

ABSTRACT

The **Sentiment Analysis with Chatbot** project is an AI-based application developed in Python that combines Natural Language Processing (NLP), speech recognition, and sentiment analysis to enable emotionally aware conversations. It detects positive, negative, or neutral sentiments in user inputs and responds empathetically while also integrating keyword-based crisis detection to trigger Twilio-powered SMS alerts for trusted contacts in high-risk situations. The system supports both voice input and text-to-speech output for natural interaction and features an accessible GUI built with CustomTkinter, offering Anonymous Mode, High Contrast Mode, and resource suggestions for inclusivity. Designed to provide real-time support, motivational guidance, and mental health resources, this chatbot delivers an intelligent, empathetic, and user-friendly solution for enhancing mental well-being.

LIST OF FIGURES

Figure 1 : Block Diagram of System Architecture

Figure 2 : GUI of chatbot running with chat box and buttons

Figure 3 : Terminal Output showing Sentiment Detection and Alert Trigger

Figure 4 : SMS Alert Received on Trusted Contact's Mobile

CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1 Project Overview	1
1.2 Project Background.....	1
1.3 Aim, Objectives, and Applications.....	2
CHAPTER 2: TECHNOLOGIES USED.....	3
2.1 Python	3
2.2 Twilio	3
2.3 Transformers.....	3
2.4 CustomTkinter.....	4
CHAPTER 3: DESIGN AND METHODOLOGY.....	5
3.1 Hardware Requirements.....	5
3.2 Software Requirements.....	5
3.3 Python Libraries and Tools	5
3.4 System Architecture	6
3.5 Methodology.....	7
3.6 Summary of Design.....	8
CHAPTER 4: IMPLEMENTATION AND RESULT.....	9
4.1 Implementation	9
4.2 Results.....	10
4.3 Visualization.....	11
4.4 Discussion	12
CHAPTER 5: CONCLUSION AND FUTURE SCOPE	13
5.1 Conclusion	13
5.2 Future Scope.....	13
REFERENCES	15

CHAPTER 1

INTRODUCTION

1.1 Project Overview

The Sentiment Analysis with Chatbot project is an Artificial Intelligence (AI)–based application that combines Natural Language Processing (NLP), speech recognition, sentiment analysis, and real-time alert systems to create an intelligent conversational agent. Unlike traditional chatbots, this system goes beyond answering queries it identifies the emotional state of the user and responds empathetically.

The chatbot processes both text and voice inputs, analyzes user emotions, and provides appropriate feedback or suggestions. In cases where critical negative sentiment or harmful expressions are detected, the system triggers an alert to a trusted contact via SMS using Twilio. The application also includes accessibility features such as Anonymous Mode, High Contrast Mode, and Text-to-Speech (TTS), making it suitable for diverse users[5][8].

1.2 Project Background

Over the past decade, chatbots have evolved from simple rule-based systems to intelligent conversational agents powered by AI. With the rise of NLP and deep learning, chatbots are now capable of understanding context, detecting emotions, and providing human-like responses. Industries such as customer service, education, and healthcare have adopted chatbot systems widely, highlighting their versatility and potential.

Mental health has become a pressing global issue, with millions of individuals experiencing stress, depression, or anxiety. However, limited access to professionals, high costs, and the stigma associated with seeking help often prevent people from getting timely support. AI-powered chatbots can bridge this gap by offering immediate, empathetic, and accessible assistance[6][7].

The Sentiment Analysis with Chatbot project builds upon these technological advancements. By integrating sentiment analysis with real-

time crisis detection and an intuitive GUI, it provides a supportive platform that can complement professional care. The project is not intended to replace mental health experts but to act as a first layer of support, raising awareness and ensuring timely intervention in critical situations.

1.3 Aim and Objectives

Aim

The primary aim of the Sentiment Analysis with Chatbot project is to design and implement an intelligent chatbot capable of analyzing user sentiment in real time, providing empathetic responses, and triggering appropriate interventions in critical situations. The system is intended to enhance accessibility, offer immediate conversational support, and act as a complementary tool to professional mental health services.

Objectives

To achieve the aim, the project is guided by the following objectives:

- Develop an AI-powered chatbot that can engage in natural conversations using both text and voice inputs.
- Integrate sentiment analysis using Natural Language Processing to classify user input as positive, negative, or neutral.
- Implement crisis detection through regex-based keyword matching to identify high-risk expressions.
- Enable real-time alerts by integrating Twilio API to send SMS notifications to trusted contacts in emergency scenarios.
- Incorporate accessibility features such as Anonymous Mode, High Contrast Mode, and Text-to-Speech for inclusive usage.
- Provide resource recommendations by linking users to professional helplines, mindfulness tools, and online support platforms.
- Test and evaluate the system's accuracy, usability, and responsiveness across different inputs and environments.

CHAPTER 2

TECHNOLOGY USED

2.1 Python

Python is a high-level, interpreted programming language widely used for Artificial Intelligence, Machine Learning, and application development. Its simplicity, readability, and vast ecosystem of libraries make it an ideal choice for building intelligent systems. In this project, Python serves as the primary programming language, enabling integration of multiple modules such as Natural Language Processing (NLP), speech recognition, sentiment analysis, and GUI development. Libraries like *Speech Recognition*, *Pytsx3*, *Transformers*, and *CustomTkinter* are leveraged to implement chatbot functionalities effectively[4].

2.2 Twilio

Twilio is a cloud communication platform that provides APIs for sending messages, making calls, and integrating real-time communication features into applications. In this project, Twilio plays a critical role in crisis management by sending SMS alerts to trusted contacts whenever harmful or suicidal expressions are detected. By using the Twilio API with Python, the system ensures immediate intervention and timely notification, thereby enhancing user safety[3].

2.3 Transformers

Transformers is an open-source library developed by Hugging Face, designed for Natural Language Processing tasks such as sentiment analysis, text classification, and conversational AI. It provides access to pre-trained models like *DistilBERT* and *BERT*, which are optimized for understanding context in human language. In this project, Transformers is used to analyze user input sentiment (positive, negative, or neutral). This allows the chatbot to generate empathetic and context-aware responses, making the interaction more meaningful and supportive[1][2].

2.4 CustomTkinter

CustomTkinter is a modern, customizable Python library built on top of Tkinter for creating graphical user interfaces (GUIs). Unlike traditional Tkinter, it offers enhanced visual elements, dark mode, themes, and improved widget design. In this project, CustomTkinter is used to design an accessible and user-friendly interface. Features such as Anonymous Mode, High Contrast Mode, Chat Window, Text Entry Box, and Resource Suggestions are implemented using CustomTkinter, ensuring that the chatbot is intuitive and inclusive for different types of users[9].

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Hardware Requirements

The proposed system requires minimal hardware resources as it is primarily a lightweight desktop-based application. The requirements are as follows:

- **Processor:** Intel i3 or higher
- **RAM:** Minimum 4 GB (8 GB recommended)
- **Storage:** 500 MB free space for Python libraries and dependencies
- **Microphone:** For speech recognition input
- **Speakers/Headphones:** For text-to-speech output
- **Internet Connection:** Required for Twilio API and Hugging Face sentiment model

3.2 Software Requirements

- **Operating System:** Windows 10 / 11 or Linux
- **Programming Language:** Python 3.8+
- **IDE/Editor:** VS Code, PyCharm, or Jupyter Notebook
- **APIs/Frameworks:** Twilio API, Hugging Face Transformers
- **Dependencies:** Installed via pip (requirements.txt)

3.3 Python Libraries and Tools

The development of *Sentiment Analysis with Chatbot* makes use of several Python libraries and tools that enable Natural Language Processing, speech interaction, and GUI design. The key libraries are:

- **CustomTkinter** – Used for creating a modern and user-friendly graphical interface with features like Anonymous Mode, High Contrast, and chat windows.
- **Threading** – Ensures smooth execution by running tasks such as voice recognition and text-to-speech in parallel with the GUI.
- **SpeechRecognition** – Converts spoken input from the user into text using Google’s Speech API[10].
- **Pytsx3** – Provides text-to-speech functionality, allowing the chatbot to respond with an audible voice[11].
- **Transformers (Hugging Face)** – Implements sentiment analysis using pre-trained NLP models to classify text as positive, negative, or neutral[5][8].
- **Twilio** – Sends SMS alerts to trusted contacts when the chatbot detects harmful or high-risk user inputs[3].

These libraries work together to create a responsive, intelligent, and accessible chatbot system capable of handling both normal conversation and emergency interventions.

3.4 System Architecture

The system follows a **modular architecture** consisting of the following components:

- **User Input Layer** – Accepts text input through the entry box or voice input via microphone.
- **Preprocessing & Sentiment Analysis** – Converts speech to text (if applicable), analyzes sentiment using Transformers.
- **Critical Keyword Detection** – Scans for predefined harmful expressions using keyword/regex matching.
- **Response Generation** – Produces chatbot responses based on sentiment classification and conversation logic.

- **Alert Module** – Triggers Twilio SMS notification to trusted contacts in crisis situations.
- **Output Layer** – Displays text in GUI chatbox and optionally provides voice response via Pyttsx3.

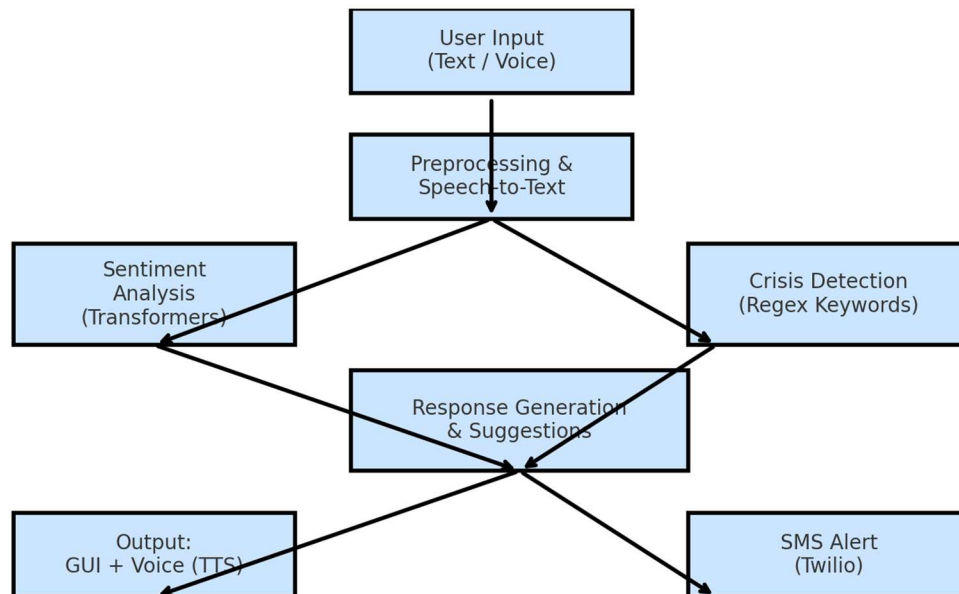


Figure 1 : Block Diagram of System Architecture

3.5 Methodology

The methodology adopted in this project can be summarized as follows:

- **Requirement Analysis** – Identification of user needs, hardware/software requirements, and feature set.
- **Data Processing** – Text and voice inputs are captured and preprocessed.
- **Sentiment Classification** – Hugging Face Transformers model classifies input into Positive, Negative, or Neutral sentiment.
- **Crisis Detection** – Regex-based keyword matching identifies harmful phrases such as “suicide,” “kill myself,” etc.

- **Alert Mechanism** – If a harmful keyword is detected, Twilio sends SMS alerts to the trusted contact[3].
- **Response Generation** – Based on sentiment, the chatbot provides empathetic, motivational, or neutral responses.
- **Accessibility Features** – Anonymous Mode, High Contrast Mode, and Text-to-Speech enhance usability.
- **Testing & Evaluation** – System performance evaluated through GUI interaction, SMS delivery confirmation, and accuracy of sentiment detection.

3.6 Summary of Design

The design of the *Sentiment Analysis with Chatbot* integrates modern AI with accessible GUI design and real-time alert systems. By combining text/voice interaction, NLP-based sentiment detection, and Twilio-powered SMS notifications, the system ensures both empathetic engagement and timely intervention. The modular architecture and lightweight hardware requirements make the application user-friendly, scalable, and adaptable for real-world use.

CHAPTER 4

IMPLEMENTATION AND RESULT

4.1 Implementation

The *Sentiment Analysis with Chatbot* system was implemented in Python, integrating multiple libraries for Natural Language Processing, speech interaction, and GUI design. The major steps in implementation are as follows:

- **GUI Development** – The interface was built using *CustomTkinter*, providing a modern layout with Anonymous Mode, High Contrast Mode, chat display, entry box, and resource buttons.
- **Speech Integration** – *SpeechRecognition* enabled voice input, while *Pytttsx3* converted chatbot responses into speech output.
- **Sentiment Analysis** – The Hugging Face *Transformers* pipeline was used for sentiment classification into Positive, Negative, or Neutral[2].
- **Crisis Detection** – Regex-based keyword matching was added to identify high-risk expressions (e.g., “suicide,” “kill myself”).
- **Alert System** – *Twilio API* was integrated to send SMS alerts to trusted contacts in case of detected harmful intent.
- **Threading** – Implemented for smooth execution of background processes like voice recognition without freezing the GUI.

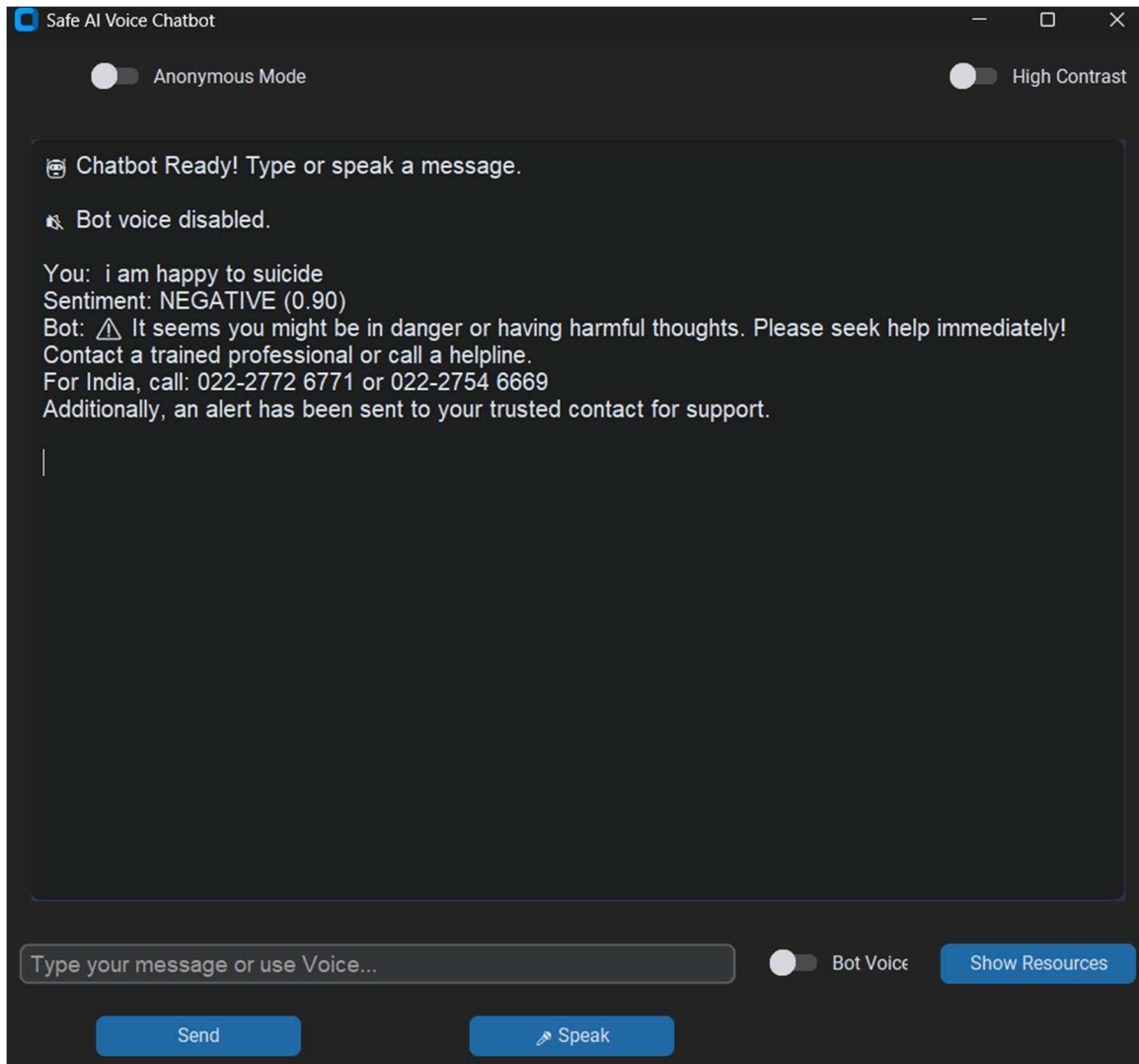


Figure 2 : GUI of chatbot running with chat box and buttons

4.2 Result

The implemented system successfully allows users to interact with the chatbot via both text and voice. Based on sentiment, it generates empathetic responses and provides helpful suggestions.

- Positive Input: The chatbot encourages and motivates the user.
- Negative Input: The chatbot provides comforting responses and directs users to resources.

- Critical Input: For harmful or suicidal keywords, the system not only provides helpline numbers but also sends an SMS alert to a trusted contact

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\siddh_ch0ux8e\modern_voice_chatbot>python -m venv venv

C:\Users\siddh_ch0ux8e\modern_voice_chatbot>venv\Scripts\activate

(venv) C:\Users\siddh_ch0ux8e\modern_voice_chatbot>python modern_chatbot6.py
Loading sentiment model...
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f (https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not recommended.
Device set to use cpu
Alert SMS sent successfully!
```

Figure 3 : Terminal Output showing Sentiment Detection and Alert Trigger

4.3 Visualization

The effectiveness of the system can be demonstrated through both textual and visual outputs:

- The GUI displays clear conversations with sentiment labels.
- The chatbot reads responses aloud if Text-to-Speech is enabled.
- When high-risk words are detected, an SMS notification is sent to the registered trusted contact.



Figure 4 : SMS Alert Received on Trusted Contact's Mobile

4.4 Discussion

The project demonstrates that integrating sentiment analysis with chatbot systems can provide not only conversational assistance but also meaningful emotional support. The results show:

- **Accurate Sentiment Detection:** The Hugging Face model effectively classifies emotions in real time.
- **Real-Time Crisis Intervention:** The Twilio-based SMS alert system ensures immediate action during high-risk conversations.
- **User Accessibility:** Features like Anonymous Mode, High Contrast Mode, and Text-to-Speech make the application inclusive for different user groups.
- **Practical Impact:** While the chatbot is not a replacement for professional therapy, it can act as a first-level support system to guide users toward appropriate help.

This validates the system's effectiveness as a mental health-oriented chatbot and highlights its potential role in preventive care[6].

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The project *Sentiment Analysis with Chatbot* successfully demonstrates the integration of Artificial Intelligence, Natural Language Processing, and real-time alert systems in building an intelligent and socially impactful chatbot. The system can identify user sentiment, generating empathetic responses, and initiating SMS alerts in case of critical or harmful inputs.

Through the combination of text-based and voice-based interactions, the chatbot provides users with an engaging and accessible experience. Features like Anonymous Mode, High Contrast Mode, and Text-to-Speech further enhance usability for diverse audiences. By linking technology with mental health support, the project highlights the potential of AI-driven tools in complementing professional care and improving accessibility to emotional assistance.

Overall, the system demonstrates that chatbots can extend beyond task automation and customer service, offering meaningful applications in the domain of mental health awareness and preventive care[12].

5.2 Future Scope

While the project achieves its core objectives, there is significant scope for future development and enhancement. Potential improvements include:

- **Conversation History:** Allow users to view or export past interactions for personal reference or clinical support.
- **User Customization:** Introduce advanced settings such as toggle switches for anonymous mode, text-to-speech, and multilingual support to personalize the chatbot experience.

- **Offline Functionality:** Implement limited offline features by caching models locally, ensuring basic sentiment detection and chatbot responses without internet dependency.
- **Feedback System:** Add a user feedback mechanism where users can rate their experience or provide suggestions for improvement, enabling iterative refinement of the system.

By implementing these additional features, the *Sentiment Analysis with Chatbot* can evolve into a more comprehensive and reliable tool, capable of addressing a broader range of user needs and expanding its practical applicability in real-world scenarios[5][7].

REFERENCES

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is All You Need*. Advances in Neural Information Processing Systems, 30.
2. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). *Transformers: State-of-the-Art Natural Language Processing*. Proceedings of EMNLP 2020: System Demonstrations, 38–45.
3. Twilio. (2023). *Twilio API for Programmable Messaging*. Retrieved from <https://www.twilio.com/docs/sms>
4. Python Software Foundation. (2023). *Python Programming Language*. Retrieved from <https://www.python.org>
5. Rao, G., Huang, W., & Lin, P. (2019). *Deep Learning for Natural Language Processing: A Review*. IEEE Access, 7, 53230–53248.
6. World Health Organization (WHO). (2022). *Mental health*. Retrieved from <https://www.who.int/health-topics/mental-health>
7. Denecke, K., Bamidis, P., Bond, C., Gabarron, E., Househ, M., Lau, A. Y., Mayer, M. A., & Merolli, M. (2015). *Recommendations for Improving Social Media for Mental Health Applications*. Studies in Health Technology and Informatics, 210, 513–517.
8. Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing* (3rd ed., Draft). Stanford University.
9. CustomTkinter. (2023). *CustomTkinter Documentation*. Retrieved from <https://github.com/TomSchimansky/CustomTkinter>
10. Anthony, L. (2020). *Speech Recognition with Python*. Real Python. Retrieved from <https://realpython.com/python-speech-recognition>
11. Pyttsx3. (2023). *Text to Speech in Python*. Retrieved from <https://pyttsx3.readthedocs.io/>

12. Miner, A. S., et al. (2016). *Smartphone-Based Conversational Agents and Responses to Questions About Mental Health, Interpersonal Violence, and Physical Health*. JAMA Internal Medicine, 176(5), 619–625.