



# **IMPACT SKILLS DEVELOPMENT PROGRAM AI & DATA SCIENCE**

## **Final Project**

**Name:**

**Faiz Ul Haq  
Sania**

**Email:**

[Ulhaqfaiz27@gmail.com](mailto:Ulhaqfaiz27@gmail.com)  
[Alisania512@gmail.com](mailto:Alisania512@gmail.com)

**GitHub Profile:**

<https://github.com/faizu125>  
<https://github.com/Sania-ali19>

**Title:**

**NLP Application Using Streamlit**

# NLP Application Using Streamlit: Comprehensive Project Report

---

## 1. Introduction

The **NLP Application Using Streamlit** represents a groundbreaking effort to simplify and streamline text analysis tasks through advanced Natural Language Processing (NLP) techniques. This project leverages cutting-edge tools and algorithms to deliver a multi-functional chatbot capable of performing various NLP tasks such as sentiment analysis, language translation, text classification, and plagiarism detection.

The chatbot's intuitive interface, powered by the Streamlit framework and the Gemini API, ensures accessibility for users with varying technical expertise. Designed with scalability and adaptability in mind, the application integrates pre-trained models and provides features like API key management and search history storage, making it suitable for professional, educational, and personal use cases.

---

## 2. Key Features

### 1. Interactive Conversations:

- Engages users with context-aware and dynamic responses, ensuring seamless interaction.

### 2. Sentiment Analysis:

- Detects the emotional tone of user input (Positive, Neutral, or Negative).
- Application: Analyzing customer feedback and social media posts.

### 3. Language Translation:

- Translates text between multiple languages with high accuracy.
  - Example: Translating “Hello” to “Hola” for English-to-Spanish conversion.
  - 4. **Text Classification:**
    - Categorizes text into predefined labels, such as spam detection or topic identification.
  - 5. **Plagiarism Detection:**
    - Uses cosine similarity to identify textual overlaps, ensuring content originality.
    - Highlights matching sections and reports a similarity percentage.
  - 6. **Search History:**
    - Stores and retrieves previous interactions, providing users with continuity and convenience.
- 

### 3. Methodology

#### Requirement Analysis and Planning

- Collaborated with stakeholders to define the project’s scope and deliverables.
- Created a roadmap to ensure timely completion of each milestone.

#### Tools and Technologies

1. **Python:** Core programming language for development.
2. **Streamlit:** Framework for building a user-friendly web interface.
3. **Hugging Face:** Pre-trained transformer models for NLP tasks.
4. **TensorFlow:** Framework for building and training machine learning models.
5. **Scikit-learn:** Lightweight library for classification and evaluation metrics.
6. **Cosine Similarity:** Algorithm for plagiarism detection.
7. **Gemini API:** Powers the chatbot for real-time query handling and response generation.

## Model Development

- Leveraged pre-trained models (e.g., BERT) for sentiment analysis and language translation.
- Built a custom cosine similarity algorithm for plagiarism detection.

## Integration and Deployment

- Integrated NLP functionalities via the Gemini API.
- Deployed using Streamlit for an interactive and dynamic user interface.

## Testing and Evaluation

- Evaluated models using accuracy, precision, recall, and F1-score.
  - Conducted user testing to ensure a smooth and intuitive user experience.
- 

# 4. Architecture

## Front-End

- Built with Streamlit to provide an interactive and accessible user experience.

## Back-End

- Python scripts handle data preprocessing, model inference, and result generation.
- API integration ensures seamless communication between components.

## Models

- Pre-trained models from Hugging Face for sentiment analysis and language translation.
- Custom-built cosine similarity algorithm for plagiarism detection.

## Deployment

- Hosted locally with a lightweight static file server.
  - Designed for future cloud deployment to enhance scalability.
- 

## 5. Challenges and Solutions

1. **Data Availability:**
    - Challenge: Limited access to domain-specific datasets.
    - Solution: Generated synthetic datasets and leveraged publicly available resources.
  2. **Integration Issues:**
    - Challenge: Conflicts between APIs and models during integration.
    - Solution: Modularized code to isolate and test each functionality independently.
  3. **Performance Optimization:**
    - Challenge: Latency due to large model sizes.
    - Solution: Optimized API calls and implemented caching mechanisms.
  4. **Scalability:**
    - Challenge: Handling concurrent user requests.
    - Solution: Designed a microservices architecture for future scalability.
- 

## 6. Future Enhancements

1. **Advanced NLP Techniques:**
  - Incorporate transformer-based models like GPT-4 for richer context understanding.
2. **Multilingual Support:**
  - Expand language translation capabilities to include more languages.
3. **Enhanced UI/UX:**

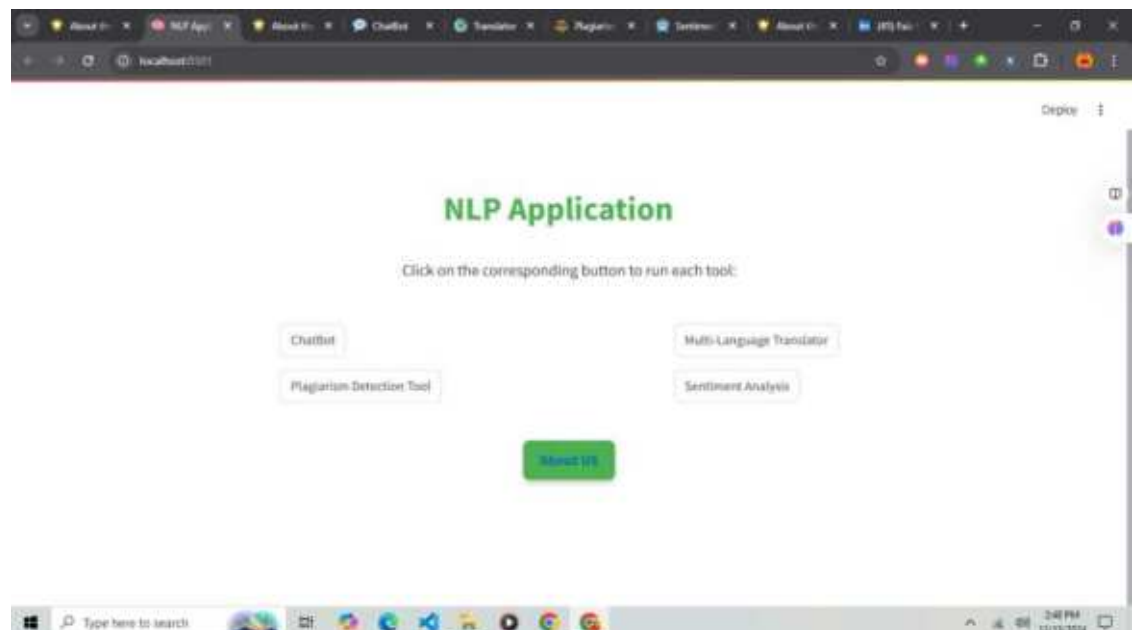
- Add voice-based interaction and visual elements for better accessibility.
4. **Additional Features:**
- Summarization for lengthy documents.
  - Sentiment trend analysis for business insights.
- 

## 7. Outcomes

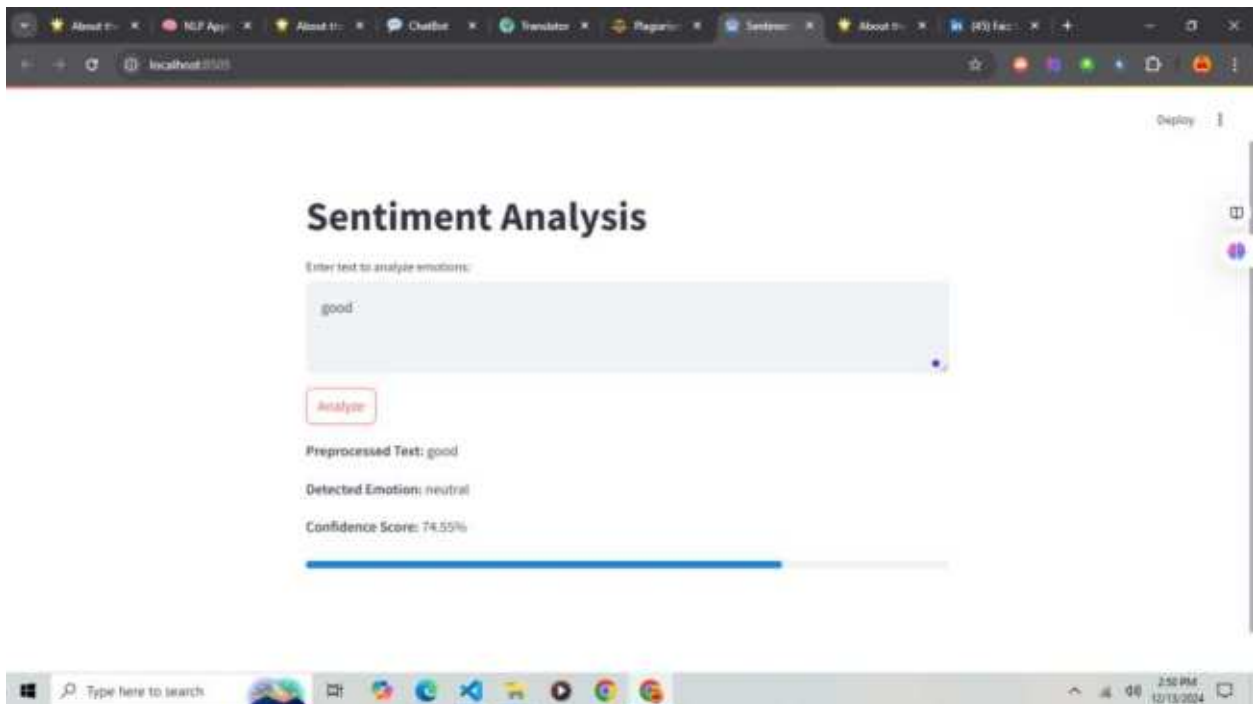
- **Deliverables:**
    - Fully functional chatbot with integrated NLP features.
    - Comprehensive project documentation and user manual.
    - Demonstration of key functionalities.
  - **Impact:**
    - Demonstrated practical applications of AI in text analysis tasks.
    - Provided a scalable foundation for future AI-driven solutions.
- 

## 8. Demonstration

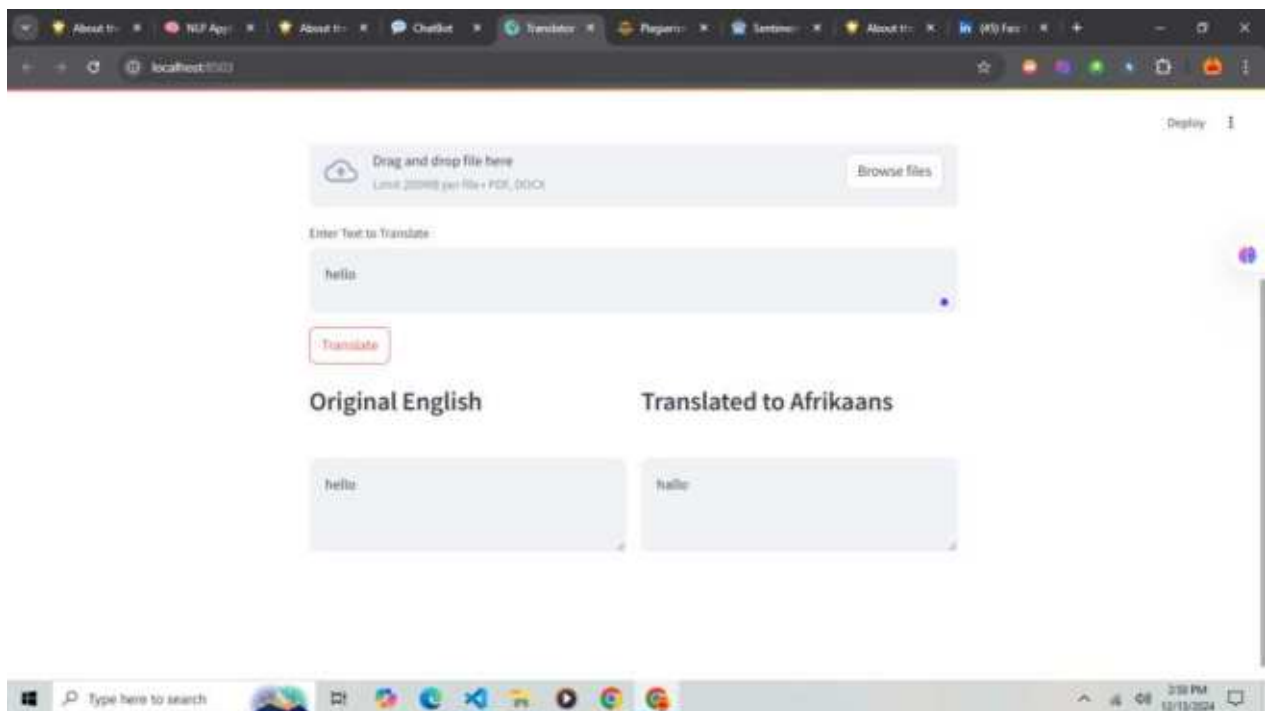
1. **Screenshots:**
- Chatbot interface



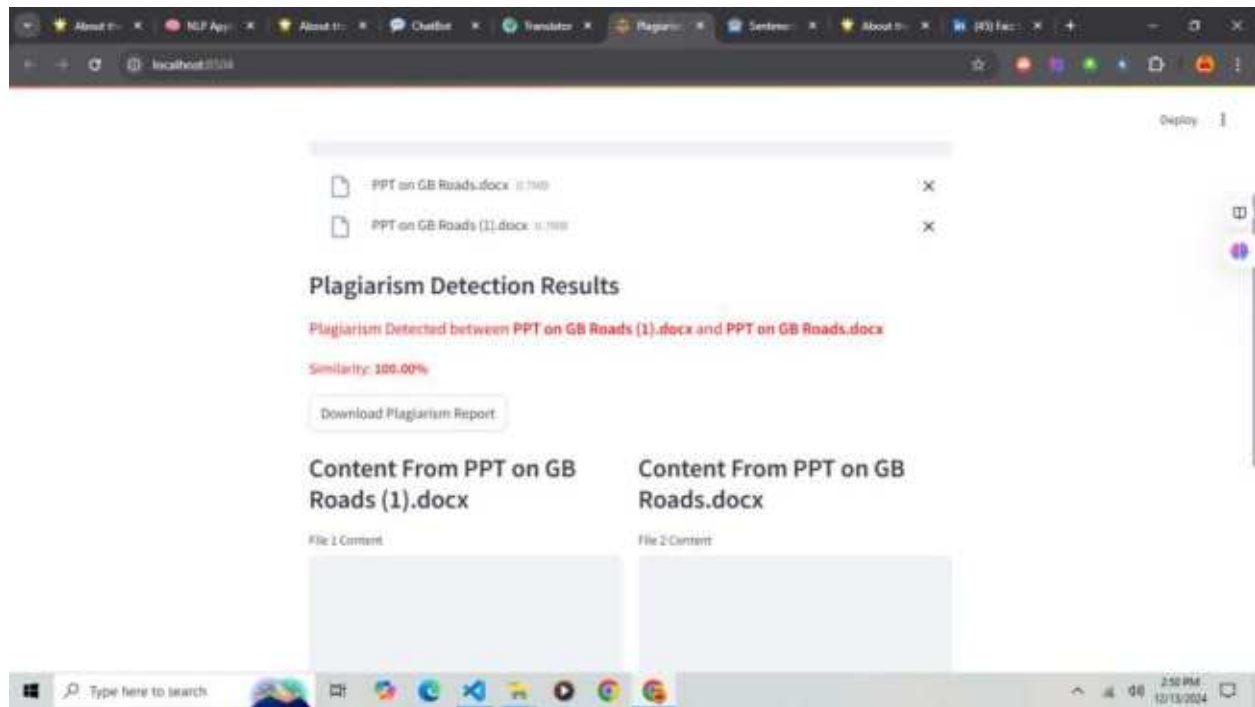
- Sentiment analysis results



- Language translation outputs



- Plagiarism detection similarity scores



## 9. Conclusion

The **NLP Application Using Streamlit** represents a significant achievement in leveraging advanced Natural Language Processing (NLP) techniques to address diverse text analysis tasks. Through meticulous development and integration of state-of-the-art tools, the project has delivered a multi-functional chatbot that is user-friendly, scalable, and versatile.

- **Key Achievements:**
  - Successfully integrated features like sentiment analysis, language translation, text classification, and plagiarism detection into a single application.
  - Overcame critical challenges in data availability, model integration, and performance optimization.



- Demonstrated the practical applications of AI, enhancing both accessibility and efficiency in language processing tasks.
-