

NLP Application using Streamlit

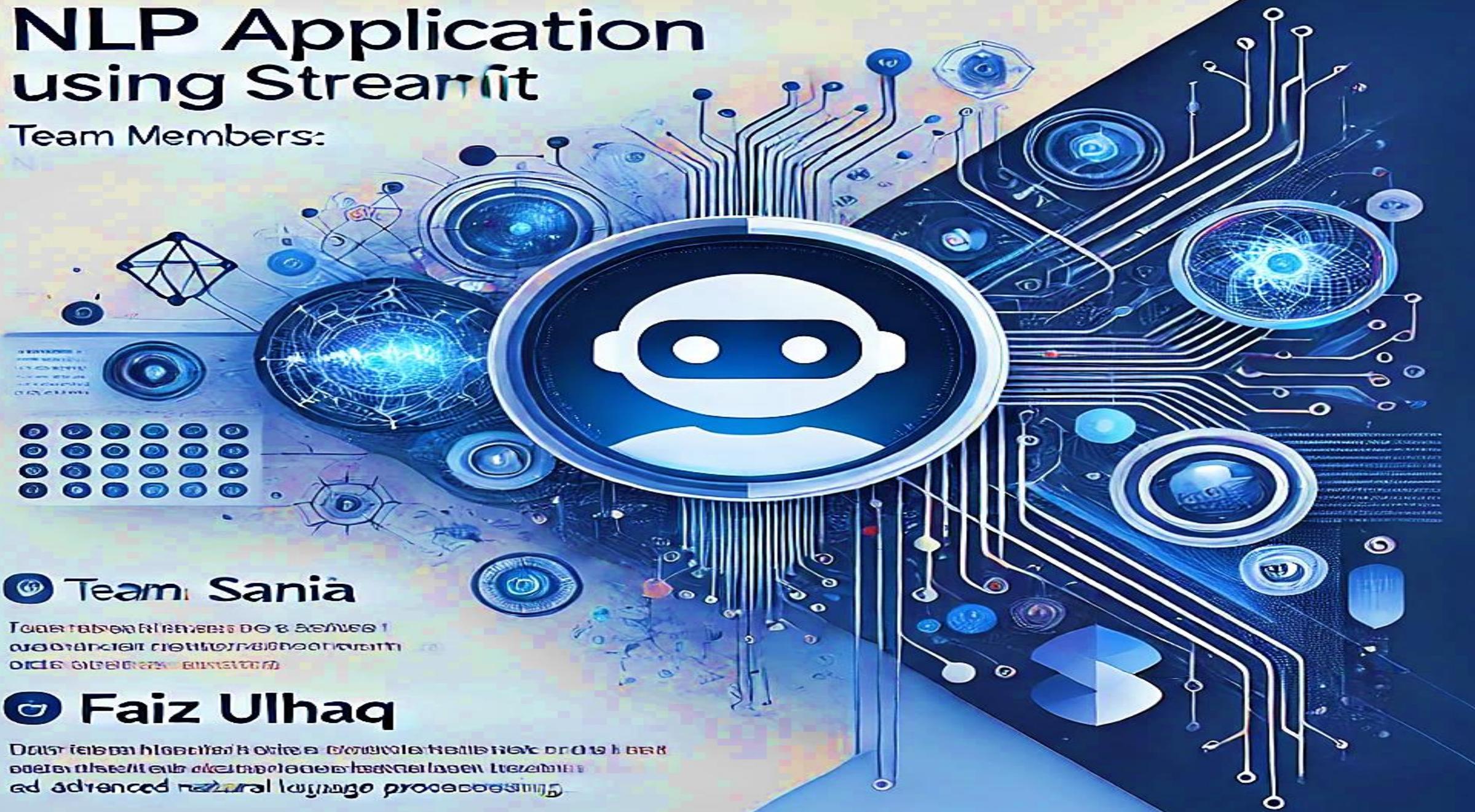
Team Members:

Team Sania

Team members: Dr. S. A. Sania
and others.

Faiz Ulhaq

Dr. Faiz Ulhaq is working on the development of NLP applications based on deep learning and advanced natural language processing.



AI Chabot Development Using Gemini API



Introduction

- The NLP Application Using Streamlit is an innovative project designed to harness the power of advanced Natural Language Processing (NLP) techniques to solve real-world text analysis problems. This app combines multiple functionalities into a user-friendly interface, making it accessible to users with varying levels of technical expertise.
- By integrating key NLP functionalities such as sentiment analysis, language translation, text classification, and plagiarism detection, this application serves as a versatile tool for handling diverse language processing tasks.

- Additionally, the app leverages pre-trained models and state-of-the-art algorithms, ensuring high accuracy and performance. The integration of a context-aware chatbot powered by the Gemini API enhances the user experience by allowing dynamic query handling and seamless interaction with the app's features.
- Designed for scalability, the app also includes advanced features like API key management and search history storage, providing users with the flexibility to adapt the system for evolving needs. This application represents a practical, multi-functional solution for professional, educational, and personal use cases in the field of NLP.

Key Features of the Chatbot:

✓ Sentiment Analysis:

- Evaluates the emotional tone of user input.
- Enables applications such as customer feedback analysis, social media monitoring, and market sentiment detection.

✓ Language Translation:

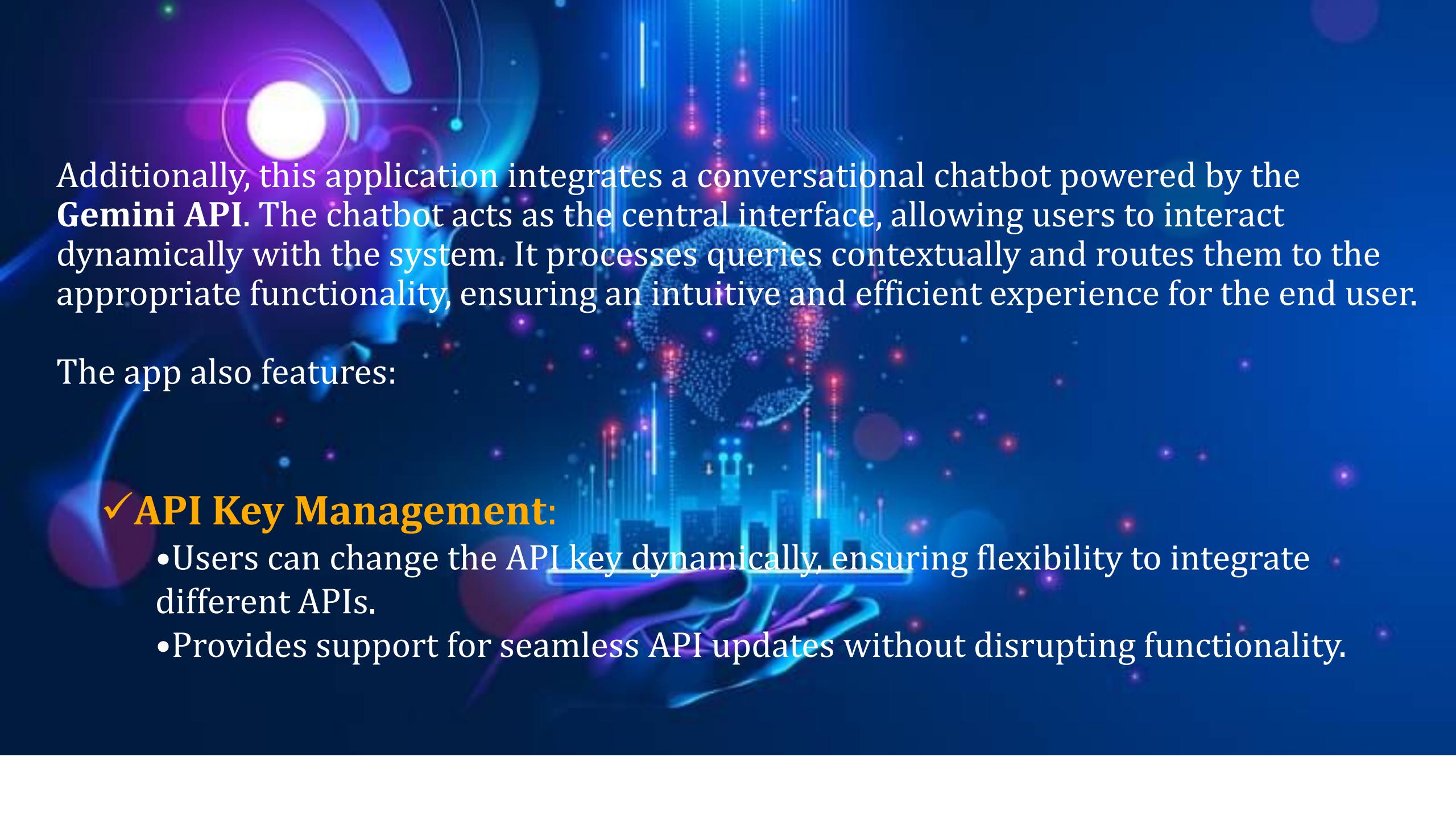
- Converts text between multiple languages seamlessly.
- Breaks down communication barriers in a multilingual environment.

✓ **Text Classification:**

- Categorizes input text into predefined labels such as spam detection, topic identification, or organizational workflows.

✓ **Plagiarism Detection:**

- Ensures the originality of content by comparing documents and identifying similarities using cosine similarity.
- Highlights matching sections and reports a similarity percentage for clarity.



Additionally, this application integrates a conversational chatbot powered by the **Gemini API**. The chatbot acts as the central interface, allowing users to interact dynamically with the system. It processes queries contextually and routes them to the appropriate functionality, ensuring an intuitive and efficient experience for the end user.

The app also features:

✓ **API Key Management:**

- Users can change the API key dynamically, ensuring flexibility to integrate different APIs.
- Provides support for seamless API updates without disrupting functionality.

✓ **Google Translate Dataset:**

- Used for training and validating the language translation feature.
- Ensures accuracy and reliability in multilingual translations.

✓ **Search and History Storage:**

- All user interactions and searches are securely stored.
- Users can access a detailed history of queries and responses, allowing easy retrieval of previous inputs and outputs.

The combination of these functionalities positions this NLP app as a comprehensive solution for diverse language processing needs, catering to both professional and educational use cases.

Objectives

✓ Develop a User-Friendly NLP App:

- Design an intuitive interface that simplifies complex NLP tasks.
- Ensure accessibility for users with varying technical expertise.

✓ Utilize State-of-the-Art Models:

- Leverage advanced machine learning models such as BERT and custom-built classifiers.
- Enhance performance and accuracy across functionalities like sentiment analysis and language translation.

✓ Ensure Scalability and Usability:

- Build a system capable of handling diverse use cases, from academic research to professional applications.
- Provide modular and extensible design to support future enhancements and increased user demand.

Scope Of Work

- **Integration of Pre-Trained Models:**

- Leverages cutting-edge pre-trained models for tasks such as sentiment analysis, language translation, and text classification.
- Ensures high performance and reduced development time by utilizing pre-built NLP frameworks.

- **Enhancing User Interaction Through Chatbot Integration:**

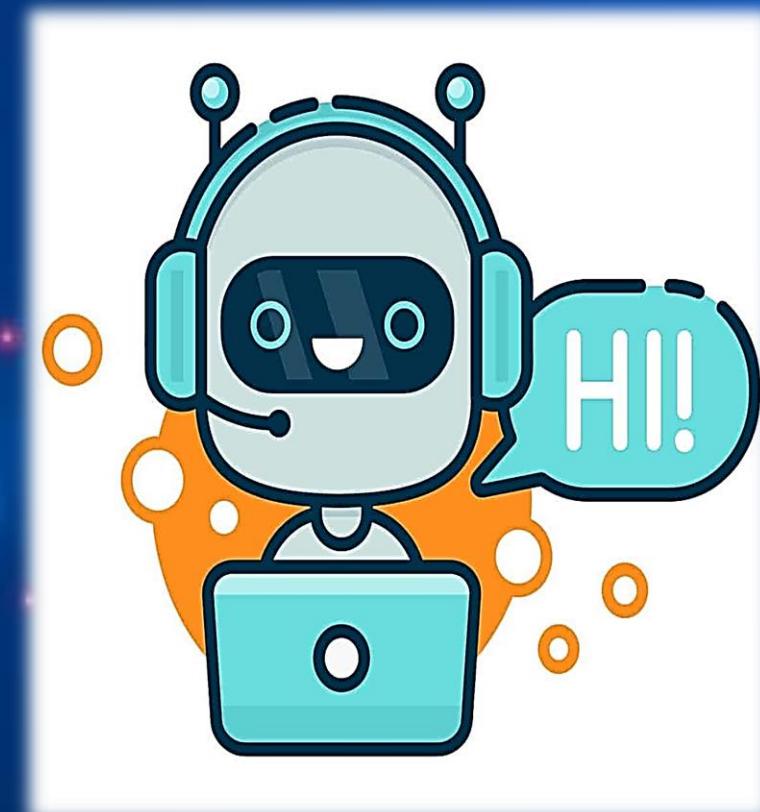
- The chatbot acts as a central interface, facilitating seamless interaction between users and NLP features.
- Ensures context-aware responses, improving the overall user experience

- 
- ✓ **Robust Data Handling and History Management:**
 - Implements storage mechanisms to log and retrieve past user interactions.
 - Allows users to access a detailed history for continuity and ease of use.
 - ✓ **Flexibility in API Management:**
 - Provides options to update and manage API keys dynamically, enabling integration of new services as needed.
 - Ensures adaptability for future enhancements and expansions.

Functionalities Overview

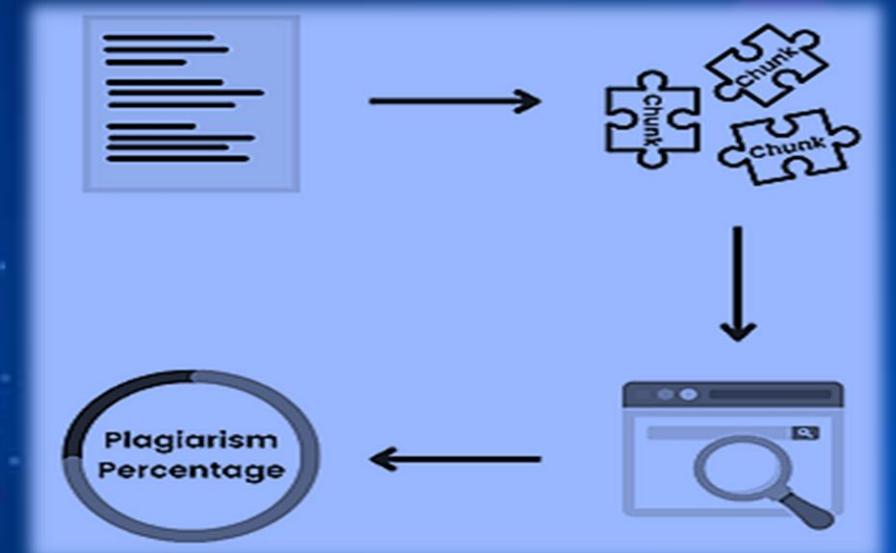
✓ Chatbot:

- The chatbot serves as the core interface for user interaction.
- It processes user queries dynamically and provides context-aware responses.
- **Example:** If a user asks, "What is the weather today?", the chatbot responds with relevant information based on integrations or pre-trained responses.



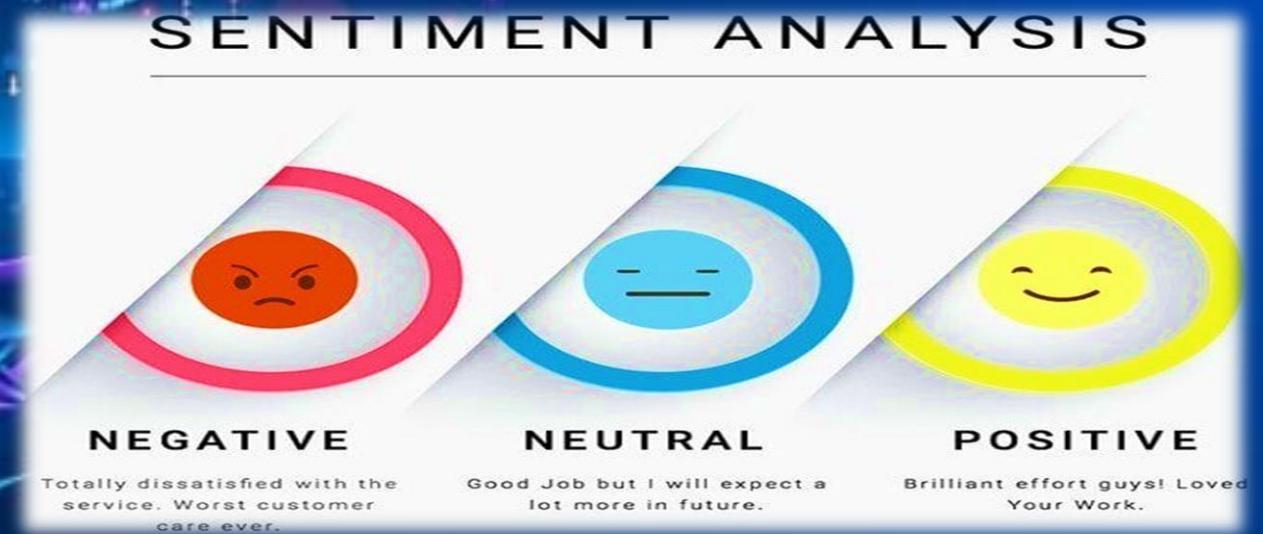
✓ Plagiarism Detection:

- Ensures originality of the content by comparing documents and highlighting similarities.
- Utilizes cosine similarity to measure the closeness of text vectors.
- **Example:** When a user submits two similar documents, the system reports a similarity score (e.g., 85%) and highlights matching sections.



✓ **Sentiment Analysis:**

- Detects the emotional tone of the user's input.
- Classifies input as positive, negative, or neutral based on predefined models.
- **Example:** For the input "I love this service!", the chatbot identifies it as positive and provides supportive feedback.



✓ Translation:

- Converts text from one language to another in real time.
- Supports multiple languages, aiding in breaking communication barriers.
- **Example:** The input "Hola" (Spanish) is translated into "Hello" (English).



Methodology

Tools and Technologies

✓ Python:

- Primary programming language used for developing the NLP functionalities.
- Offers extensive libraries and frameworks for machine learning and text processing.



✓ **Streamlit:**

- Framework for building the user-friendly interface.
- Provides interactive and dynamic web applications for showcasing NLP features.

✓ **Hugging Face:**

- Used for pre-trained transformer models like BERT for tasks such as sentiment analysis and language understanding.
- Facilitates quick implementation and fine-tuning of state-of-the-art models.



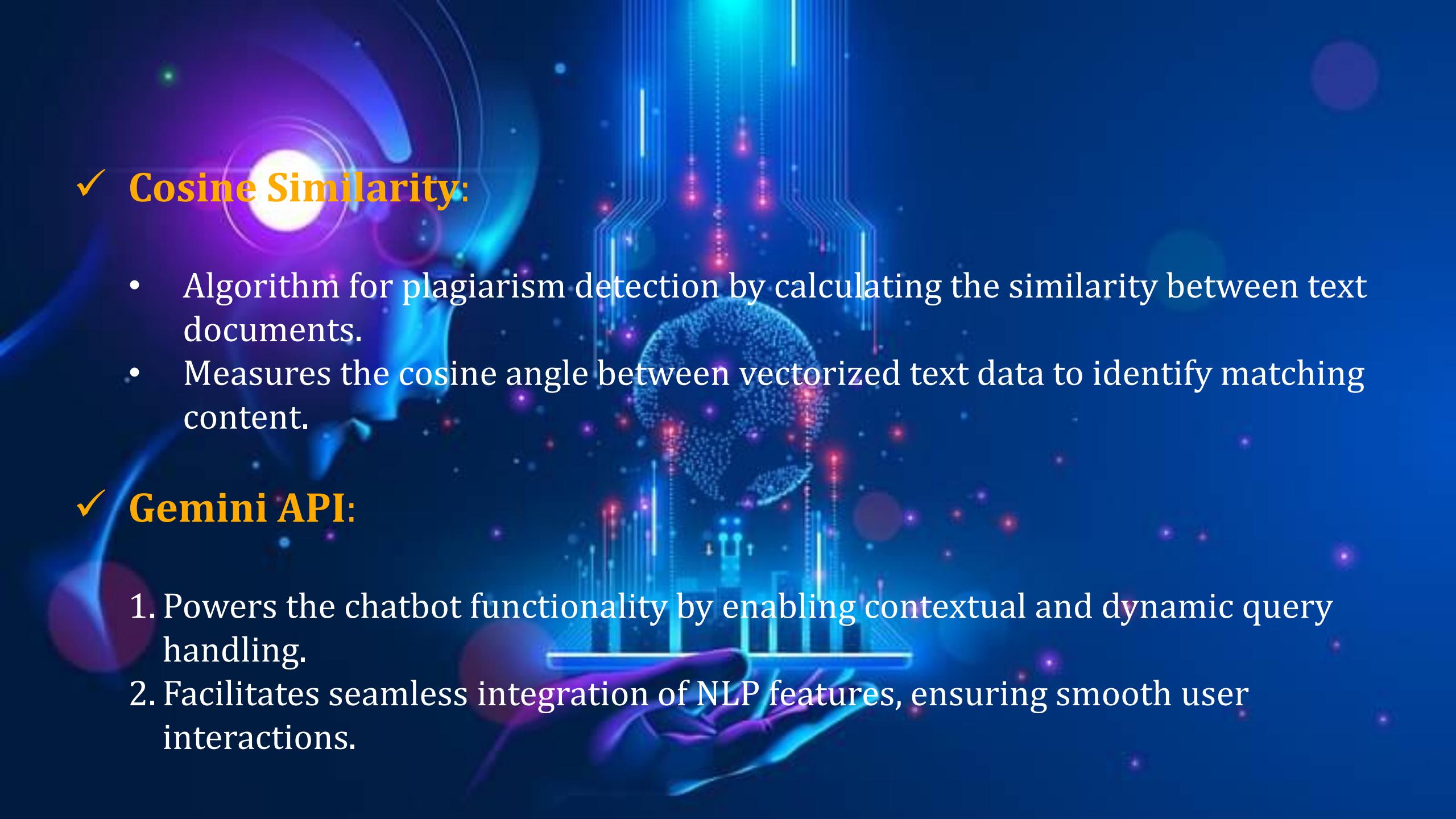
✓ **TensorFlow:**

- Framework for developing and training machine learning models.
- Optimized for handling large datasets and performing complex computations efficiently.

✓ **Scikit-learn:**

- Supports machine learning algorithms for text classification and evaluation metrics.
- Lightweight and easy to integrate for specific tasks like classification.





✓ **Cosine Similarity:**

- Algorithm for plagiarism detection by calculating the similarity between text documents.
- Measures the cosine angle between vectorized text data to identify matching content.

✓ **Gemini API:**

1. Powers the chatbot functionality by enabling contextual and dynamic query handling.
2. Facilitates seamless integration of NLP features, ensuring smooth user interactions.

Integration and Deployment

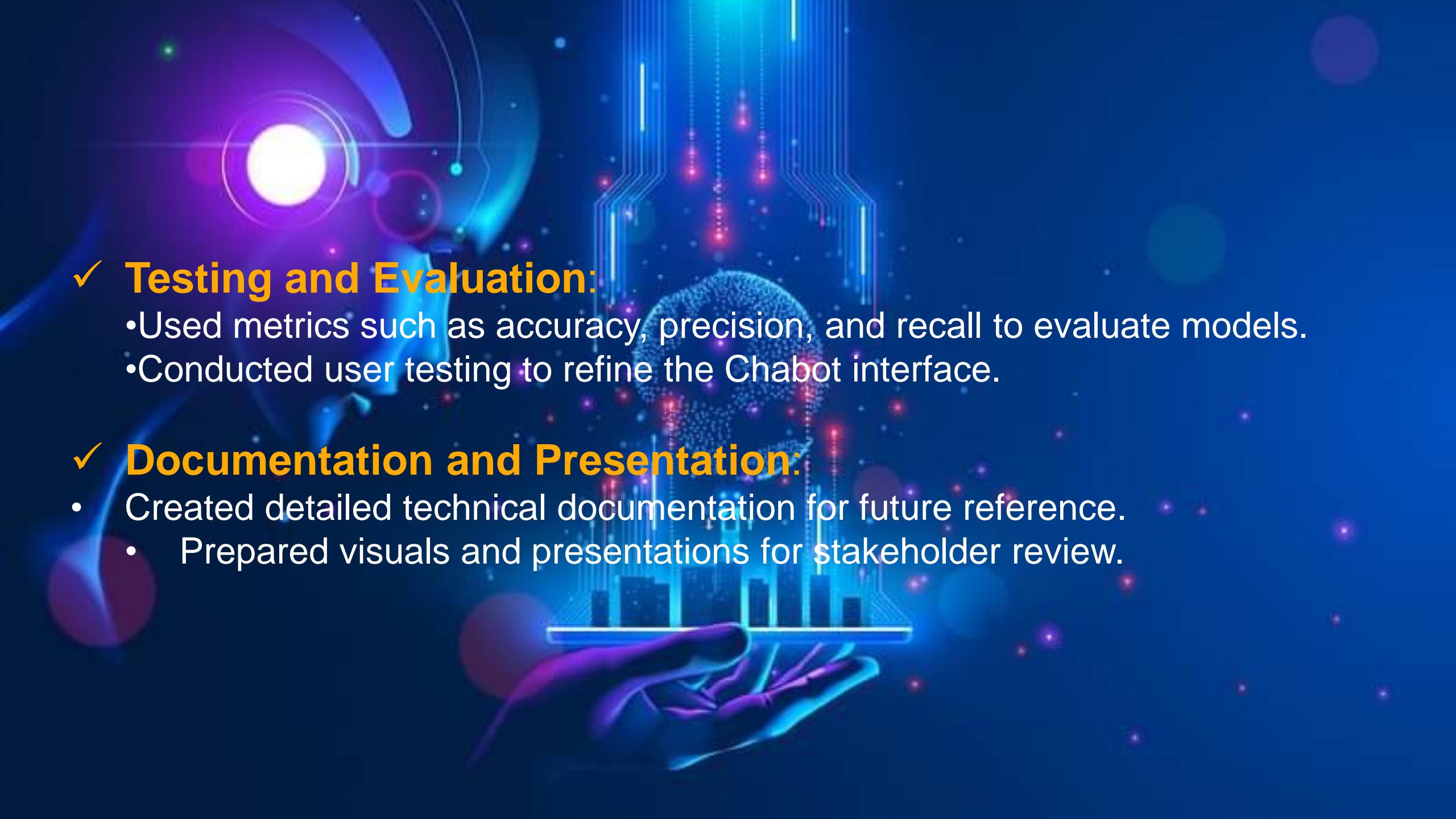
✓ Requirement Analysis and Planning:

- Collaborated with stakeholders to define the scope and features.
- Outlined key deliverables and created a development roadmap.

✓ Data Collection and Preprocessing:

- Sourced datasets from platforms like Kaggle and OpenAI Datasets.
- Preprocessing steps included tokenization, removing stopwords, and stemming for cleaner input data.

- 
- ✓ **Model Development:**
Leveraged pre-trained models like BERT for sentiment analysis.
Developed custom models for plagiarism detection using cosine similarity.
 - ✓ **Integration and Deployment:**
Integrated functionalities via Gemini API to ensure modular and scalable code.
Deployed the app using Streamlit, allowing user-friendly interactions.



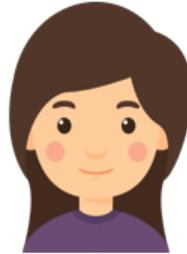
✓ **Testing and Evaluation:**

- Used metrics such as accuracy, precision, and recall to evaluate models.
- Conducted user testing to refine the Chabot interface.

✓ **Documentation and Presentation:**

- Created detailed technical documentation for future reference.
 - Prepared visuals and presentations for stakeholder review.

Chatbot
Designer

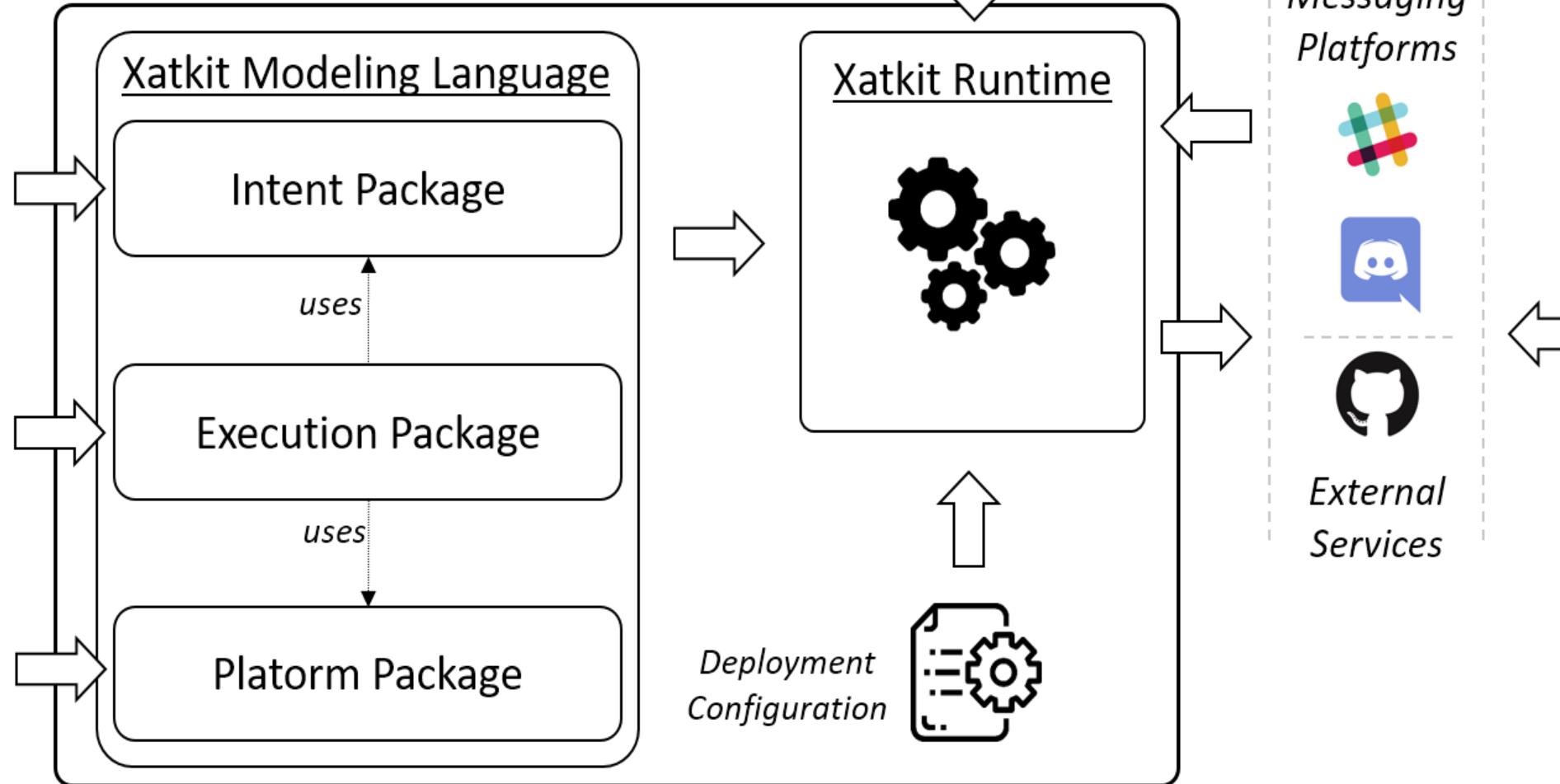


Platform
Designer



*Platform-independent
chatbot definition*

*Intent Recognition Providers
(platform-specific)*



Chatbot
User



Architecture

✓ Front-End:

- Developed with Streamlit for interactive user experiences.
- Users can input text, view results instantly, and interact with various features.

✓ Back-End:

- Utilizes Gemini API for robust integration of NLP tasks.
- Python scripts manage preprocessing, model inference, and result generation.



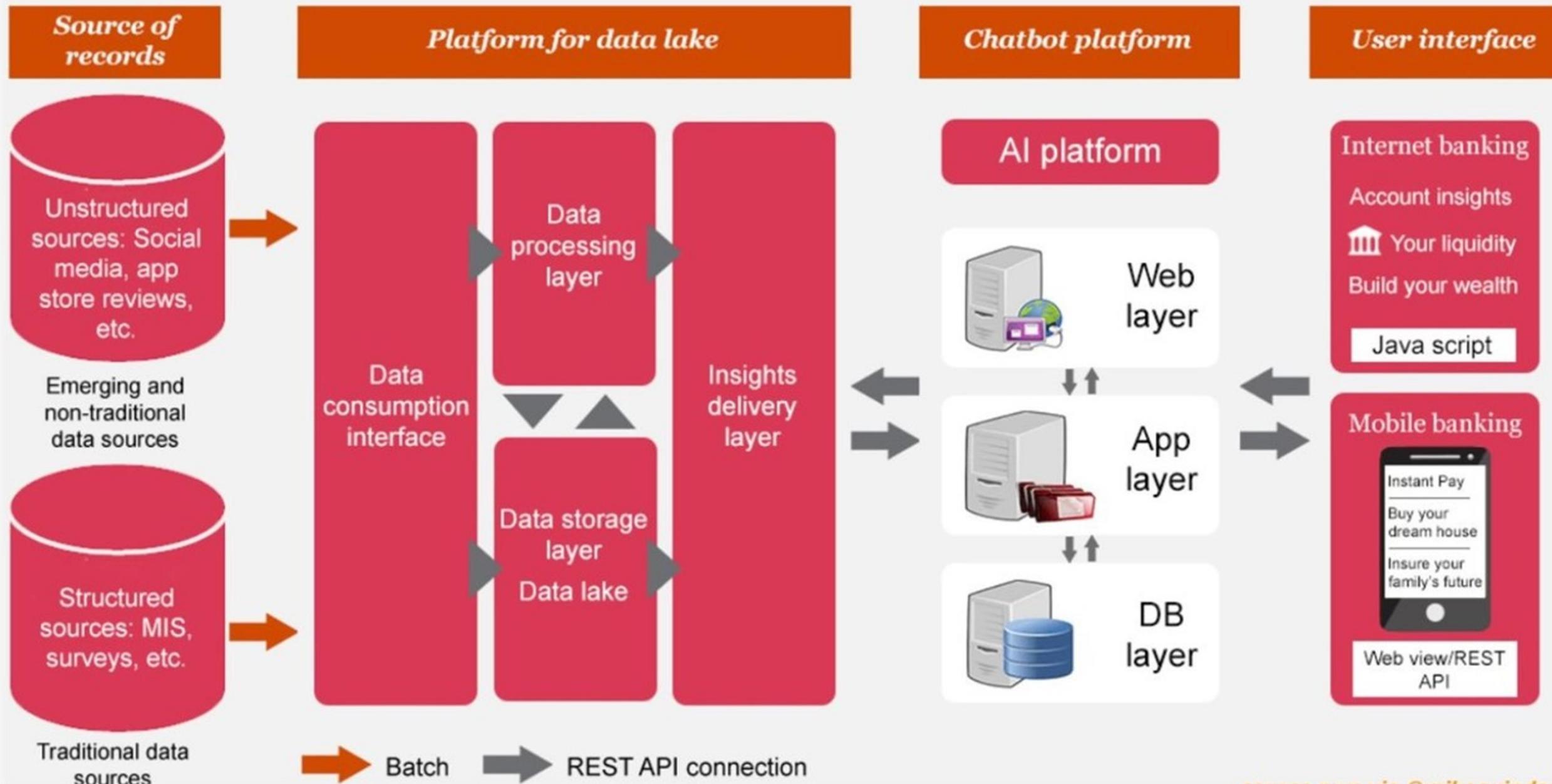
✓ **Models:**

- Hugging Face models for NLP tasks like sentiment analysis and translation.
- Custom cosine similarity algorithm for plagiarism detection.

✓ **Deployment:**

- Hosted locally with a lightweight static file server for testing.
- Future-ready for cloud deployment to enhance scalability.

Typical architecture of the chatbot platform



Challenges Faced

✓ Data Availability:

- Limited access to domain-specific datasets.
- **Solution:** Generated synthetic datasets to simulate real-world scenarios.

✓ Integration Issues:

- APIs and models sometimes conflicted during integration.
- **Solution:** Adopted a modular design, isolating each functionality for independent testing.





✓ **Performance Optimization:**

- Latency issues due to large model sizes.
- **Solution:** Optimized API calls and implemented caching mechanisms.

✓ **Scalability:**

- The chatbot needed to handle concurrent users effectively.
- **Solution:** Designed with microservices in mind for future cloud-based scaling.

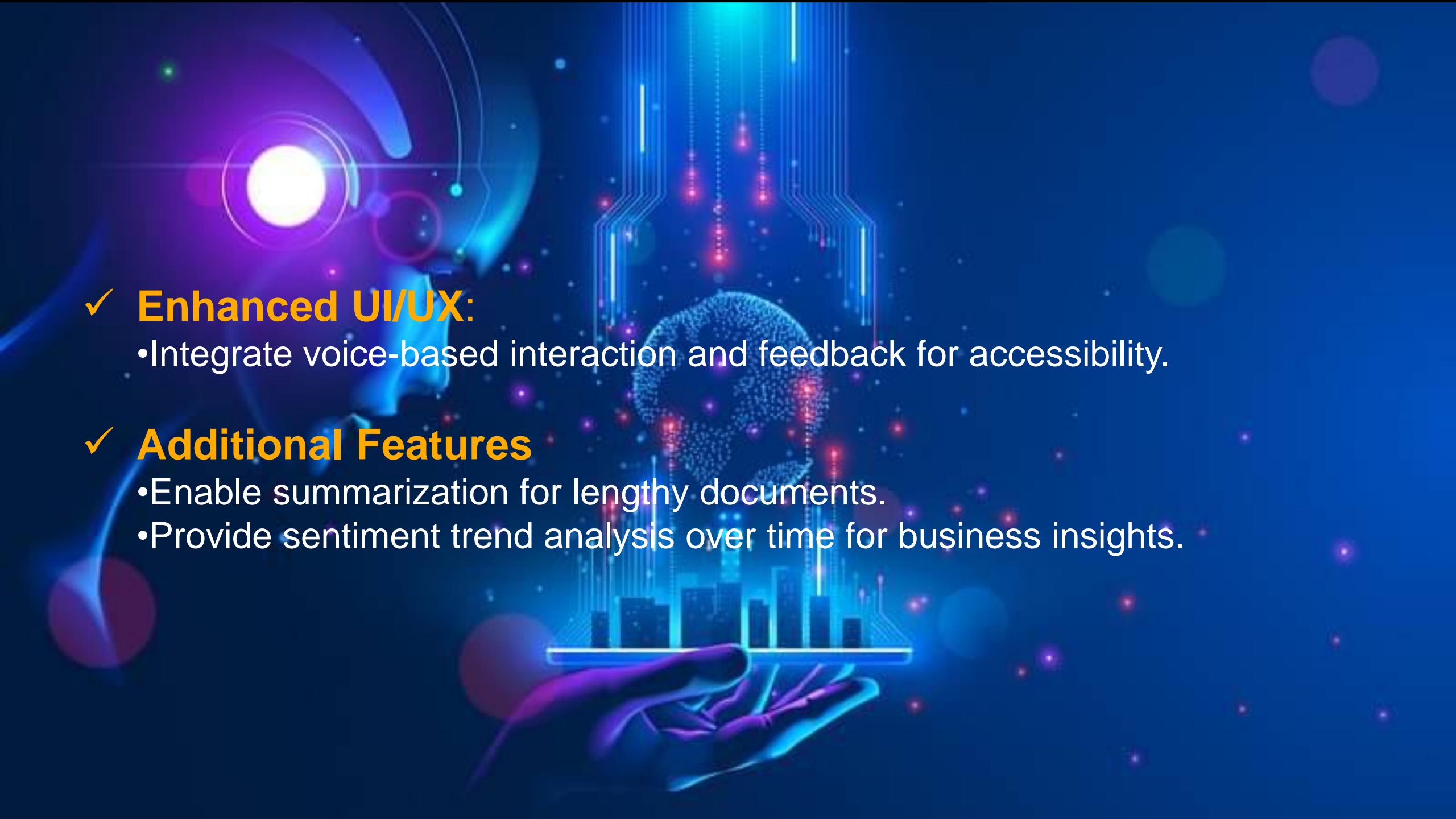
Future Enhancements

✓ Advanced NLP Techniques:

- Incorporate transformer-based models like GPT-4 for richer context understanding.

✓ Multilingual Support:

- Expand to support languages like Mandarin, Arabic, and Hindi for global reach.

- 
- ✓ **Enhanced UI/UX:**
 - Integrate voice-based interaction and feedback for accessibility.
 - ✓ **Additional Features**
 - Enable summarization for lengthy documents.
 - Provide sentiment trend analysis over time for business insights.

Outcomes

✓ Deliverables:

- Fully functional chatbot with integrated NLP features.
- Detailed documentation covering development, testing, and user interaction for future reference.
- Demonstration of key features, including the chatbot's capabilities in handling sentiment analysis, translation, classification, and plagiarism detection.



✓ Impact:

- Demonstrated practical applications of AI in everyday tasks.
- Served as a foundational project for advanced AI solutions, providing insights for future enhancements and scalability.



✓ **Budget and Resources:**

- **Tools:** Included free and subscription-based APIs to support diverse functionalities.
- **Hardware:** Required high-performance systems to efficiently manage computational demands.

Demonstration

A screenshot of a web browser window titled "NLP Application". The browser has multiple tabs open, including "About the", "NLP Appli", "ChatBot", "Translator", "Plagiarism", "Sentiment", and "About the". The main content area displays the "NLP Application" title and a message: "Click on the corresponding button to run each tool:". Below this, four buttons are visible: "ChatBot", "Plagiarism Detection Tool", "Multi-Language Translator", and "Sentiment Analysis". A green button labeled "About US" is also present. The browser's address bar shows "localhost:8501". The taskbar at the bottom includes icons for File Explorer, Edge, and other applications.

NLP Application

Click on the corresponding button to run each tool:

ChatBot

Plagiarism Detection Tool

Multi-Language Translator

Sentiment Analysis

About US

Type here to search

2:48 PM
12/13/2024

Meet the Developers



Faiz Ul Haq

Frontend & NLP Expert

[GitHub](#)

[LinkedIn](#)

[Fiverr](#)

[Upwork](#)



Sania Ali

Backend & ML Specialist

[GitHub](#)

[LinkedIn](#)

[Fiverr](#)

[Upwork](#)

Deploy

API Key Management

Enter New API Key

AlzaSyB-b8Vale92mZL7RsvlQEts



Update API Key

New Chat

History



Delete hello

LLM Bot Using Gemini API

Welcome! Ask me anything.

USER: hello

BOT: Hello there! How can I help you today?

USER: hello

BOT: BOT: Hello again! Is there something specific you need assistance with?

USER:

Type your message here...



Type here to search



Multi-Language Text Translator



Translate text between multiple languages with ease! Select the source and target languages by their names, input the text, and get your translation instantly.

Select Source Language

English

Select Target Language

Afrikaans

Upload a Word or PDF file (Optional)



Drag and drop file here

Limit 200MB per file • PDF, DOCX

Browse files

Enter Text to Translate

Translate

Plagiarism Detection Tool

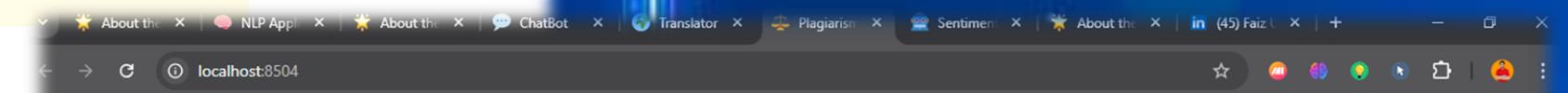
Upload at least two files (Word, PDF, or TXT). The tool will compare all files to Detect Plagiarism.

Choose PDF, DOCX, or TXT files

Drag and drop files here
Limit 200MB per file • PDF, DOCX, TXT

Browse files

Please upload at least two files for plagiarism detection.



PPT on GB Roads.docx 0.7MB

PPT on GB Roads (1).docx 0.7MB

Plagiarism Detection Results

Plagiarism Detected between PPT on GB Roads (1).docx and PPT on GB Roads.docx

Similarity: 100.00%

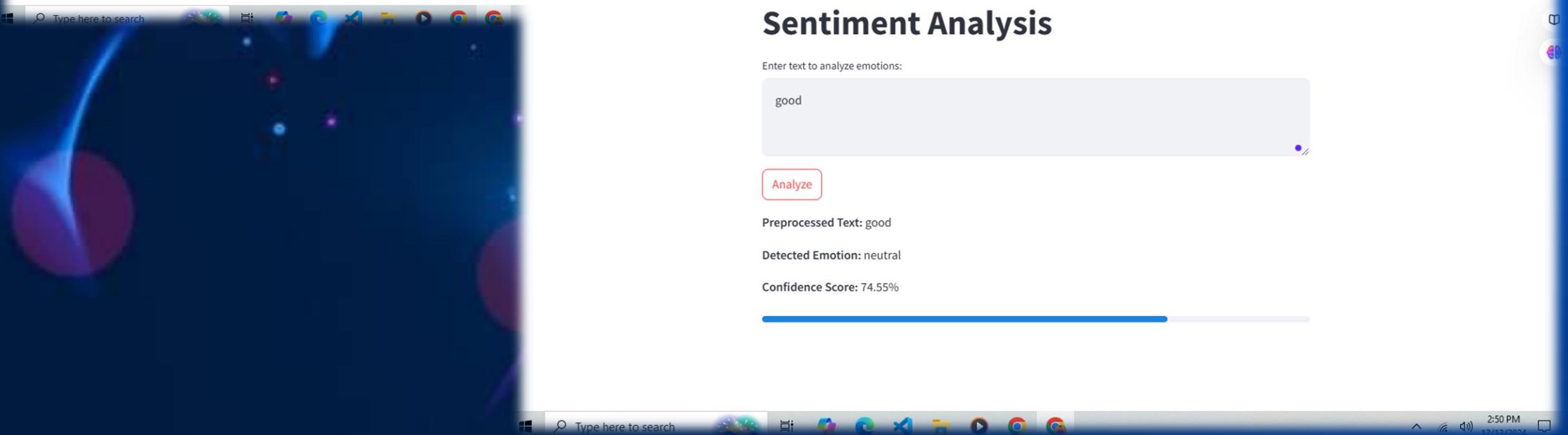
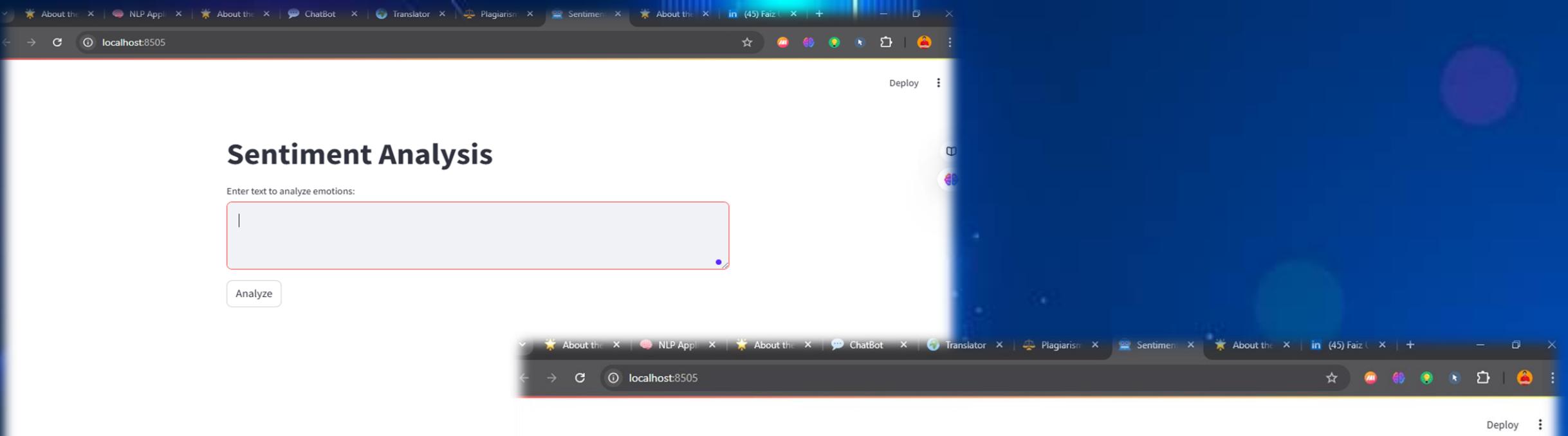
Download Plagiarism Report

Content From PPT on GB Roads (1).docx

File 1 Content

Content From PPT on GB Roads.docx

File 2 Content





Deploy :



Drag and drop file here

Limit 200MB per file • PDF, DOCX

Browse files

Enter Text to Translate

hello

Translate

Original English

hello

Translated to Afrikaans

hallo



Type here to search





Deploy

NLP Application

Click on the corresponding button to run each tool:

[ChatBo](#)[Multi-Language Translator](#)[Plagiarism Detection Tool](#)[Sentiment Analysis](#)[About US](#)

Type here to search

9:04 AM
12/30/2024

Conclusion

- This slide highlights the achievements of the project, showcasing the development of a versatile chatbot capable of performing multiple NLP tasks.
- It emphasizes the ability to overcome challenges like API integration, performance bottlenecks, and scalability concerns.
- The project demonstrates how AI can simplify complex tasks such as sentiment analysis, translation, and plagiarism detection in practical, user-friendly ways.



ANY QUESTIONS?

Adobe Stock / PAVLONOV



THANK YOU FOR
YOUR ATTENTION