**1. Introduction**

The **Sentiment Analysis Tool** is a web-based application designed to help users analyze the sentiment of text data. Built using **Django** and **Transformers' pre-trained models**, this tool provides an AI-powered sentiment analysis service with a simple and user-friendly interface.

The tool categorizes text into three sentiment labels: **positive**, **neutral**, and **negative**, based on its content.

This project also includes a **Privacy Policy** page and allows for **voice note transcription** to text using **Google Speech Recognition**.

**2. Features**

* **Sentiment Analysis**: Analyze the sentiment of any given text (positive, neutral, or negative).
* **Voice Note Transcription**: Upload or record a voice note and get a transcription in text format.
* **Privacy Policy**: A dedicated page for the privacy policy outlining the data collection process.
* **User-friendly Interface**: A simple and intuitive design for the users to easily interact with the tool.
* **JSON Response**: Receive analysis results in JSON format, with sentiment and confidence score.

**3. Technologies Used**

* **Django**: Backend framework for building the web application.
* **Transformers (Huggingface)**: Provides pre-trained models for sentiment analysis using **DistilBERT**.
* **SpeechRecognition**: Python library for converting speech into text via **Google Web Speech API**.
* **HTML/CSS**: Frontend technologies for building the user interface.
* **JavaScript (AJAX)**: To handle dynamic requests to the backend for sentiment analysis.

**4. Installation and Setup**

**Prerequisites**

Before setting up the project, ensure you have the following installed on your system:

* Python 3.x
* Django 3.x or later
* pip (Python package installer)
* A virtual environment (optional, but recommended)

**5. Usage**

**Sentiment Analysis**

1. **Text Input**: Users can enter any text into the provided textarea on the Analyze Page and click the "Analyze" button to get the sentiment analysis result.
2. **Voice Notes**: Users can record or upload voice notes which will be transcribed to text, then analyzed for sentiment.

**Example Workflow**:

* Navigate to the **Analyze Page** from the home page.
* Enter text or upload a voice note.
* Click on **Analyze** to see the sentiment and confidence percentage.

**Privacy Policy**

* Users can view the **Privacy Policy** by clicking the **Privacy Policy** link in the navigation bar.
* The policy outlines how user data is collected and handled.

**6. File Structure**

**grant\_writing\_tool/**

**│**

**├── app/ # Main application directory**

**│ ├── migrations/ # Database migrations**

**│ ├── \_\_init\_\_.py # Initialization file**

**│ ├── admin.py # Admin configurations**

**│ ├── apps.py # Application settings**

**│ ├── models.py # Database models (if applicable)**

**│ ├── views.py # Views for handling requests**

**│ ├── urls.py # URLs routing**

**│ └── templates/ # HTML templates**

**│ └── index.html # Home page template**

**│ └── analyze.html # Sentiment analysis page template**

**│ └── privacy\_policy.html # Privacy policy page template**

**│**

**├── db.sqlite3 # SQLite database**

**├── manage.py # Django management script**

**├── requirements.txt # List of dependencies**

**└── venv/ # Virtual environment directory**

**7. Privacy Policy**

**Data Collection**

We collect the following data from users:

* **Text Input**: The text entered for sentiment analysis.
* **Voice Notes**: If a user uploads a voice note, it is transcribed to text and analyzed.

**Data Usage**

The collected data is used solely for the purpose of providing sentiment analysis and improving the user experience. We do not share user data with third parties.

**Data Security**

All data is processed securely, and we take necessary measures to protect it from unauthorized access.

**10. Contributing**

We welcome contributions to this project! If you have any improvements, bug fixes, or suggestions, please follow the steps below:

1. Fork the repository.
2. Create a new branch for your changes.
3. Make your changes and commit them with a descriptive message.
4. Push your changes to your forked repository.
5. Open a pull request.