****

**Name: Ahsan Ali**

**Roll No: 061**

**Task: 13**

**Section: BSAI-4B**

**Subject: PAI Lab**

**Submitted To: Sir Rasikh**

**Title:**

**Keyboard Auto Suggest NLP**

**Project Overview:**

This project is a web-based Natural Language Processing (NLP) tool developed using Python and Flask. It helps users correct grammatically incorrect or incomplete sentences, auto-complete them using a language model, and count the number of corrections made.

**Objective:**

To design and implement a smart application that:

- Detects and corrects grammar mistakes in user-entered text.

- Completes incomplete or partial sentences in a meaningful way.

- Counts and displays the number of corrections made.

**Technologies Used:**

- Python – Backend logic and processing

- Flask – Lightweight web framework

- TextBlob – Grammar and spelling correction

- HuggingFace Transformers (GPT-2) – For sentence completion

- difflib – To count and list changes between original and corrected text

**Tools & Libraries:**

- Flask

- TextBlob

- Transformers

- Torch

- HTML/CSS for frontend

**Features Implemented:**

- Web interface for text input

- Text correction using TextBlob

- Sentence generation using GPT-2

- Correction counter using difflib

- Clear results display with original, corrected, completed text, and number of changes

**Application Flow:**

1. User inputs a grammatically incorrect or incomplete sentence.

2. Backend corrects the text using TextBlob.

3. GPT-2 predicts and completes the corrected sentence.

4. A difference checker lists the number of corrections.

5. The results are displayed back to the user.

**Folder Structure:**

/grammar-corrector-app

│

├── app.py

├── templates/

│ ├── index.html

│ └── result.html

├── static/

│ └── style.css

├── requirements.txt

└── README.md

**Example:**

Input: he go to store but not buy anything

Corrected: He went to the store but not buy anything.

Completed: He went to the store but didn’t buy anything because he forgot his wallet.

Corrections Made: 3

**Setup Instructions:**

Install dependencies:

pip install -r requirements.txt

**Run the app:**

python app.py

Open browser at `http://127.0.0.1:5000/`

**Screenshots:**

**Code: app.py**

from flask import Flask, render\_template, request

from textblob import TextBlob

from transformers import pipeline, set\_seed

import difflib

app = Flask(\_\_name\_\_)

# Load a transformer pipeline for sentence completion (GPT-2)

text\_generator = pipeline("text-generation", model="gpt2")

set\_seed(42)

def get\_corrections(original\_text):

    blob = TextBlob(original\_text)

    corrected\_text = str(blob.correct())

    # Count differences

    diff = list(difflib.ndiff(original\_text.split(), corrected\_text.split()))

    changes = [word for word in diff if word.startswith('- ') or word.startswith('+ ')]

    num\_corrections = len([c for c in changes if c.startswith('- ')])

    return corrected\_text, num\_corrections, changes

def complete\_sentence(text):

    # Ensure the input is not too long

    prompt = text.strip()

    if not prompt.endswith(('.', '?', '!')):

        prompt += '.'

    generated = text\_generator(prompt, max\_length=50, num\_return\_sequences=1)

    completed\_text = generated[0]['generated\_text']

    return completed\_text

@app.route('/')

def index():

    return render\_template('index.html')

@app.route('/process', methods=['POST'])

def process():

    user\_input = request.form['user\_text']

    corrected\_text, num\_corrections, changes = get\_corrections(user\_input)

    completed\_text = complete\_sentence(corrected\_text)

    return render\_template(

        'result.html',

        original=user\_input,

        corrected=corrected\_text,

        completed=completed\_text,

        num\_corrections=num\_corrections,

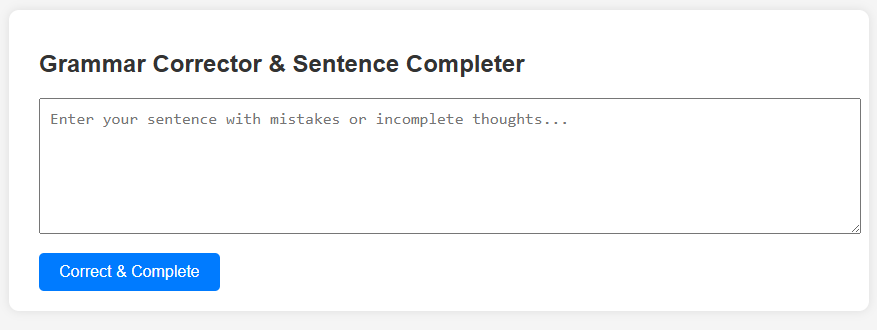
        changes=changes

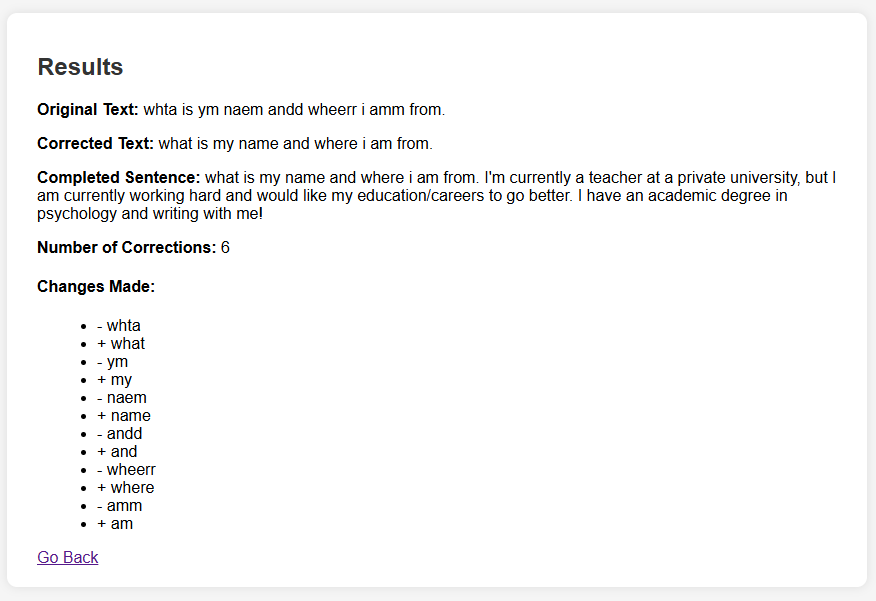
    )

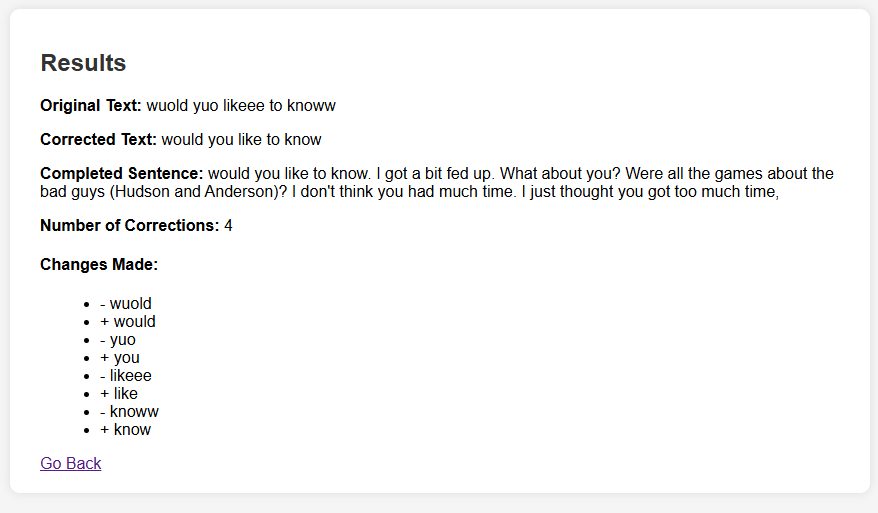
if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**Output:**

****

****

****

**Conclusion:**

This project demonstrates how simple NLP tools can be integrated into web apps to enhance grammar and sentence structure automatically. It provides a hands-on learning experience in text processing, Flask integration, and working with transformer models.