Spelling Checker Application Documentation

Overview

This document provides detailed documentation for the Spelling Checker application built using Python.

The application uses the TextBlob library for spelling correction and Tkinter for creating a graphical user interface.

It allows users to enter incorrectly spelled text and receive corrected output instantly.

Project Objective

The objective of this project is to demonstrate how natural language processing and GUI programming can be combined to build a simple and useful desktop application.

Technologies Used

Python

Used as the core programming language.

Tkinter

Provides the graphical user interface components.

TextBlob

Used to perform spelling correction using NLP techniques.

Application Architecture

The application follows an event driven architecture.

User actions such as button clicks trigger Python functions that perform the required processing.

Code Explanation

Import Statements

TextBlob is imported for spelling correction and Tkinter is imported for GUI development.

Spelling Correction Function

The correct_spelling function retrieves text from the input field, applies spelling correction,

and updates the output field.

Main Window Function

The main_window function initializes the window, configures UI components,

and starts the GUI event loop.

Global Variables

Entry widgets are declared global to allow access inside the spelling correction function.

User Interface Components

Labels

Labels guide the user on where to input text and view results.

Entry Fields

Two input fields are used for incorrect and corrected spelling.

Button

The button triggers the spelling correction process.

Application Workflow

1 User enters incorrect spelling.

2 User clicks the Check button.

3 Spelling correction logic executes.

4 Corrected text is displayed to the user.

Limitations

Only basic spelling correction is supported.

Grammar checking and suggestions are not included.

Future Enhancements

Add multi sentence correction.

Improve UI styling.

Add suggestion based corrections.

Conclusion

This project demonstrates how Python can be used to create desktop applications by combining GUI programming with natural language processing.

Author

Satyam Gajjar