**Task-3**

**Introduction**The provided Python script implements a graphical user interface (GUI) application using tkinter for checking spelling errors in user input. It leverages the SpellChecker module from the pyspellchecker library to identify and suggest corrections for misspelled words. This application is designed to notify the user when one or more incorrect words are entered, offering suggestions for correction and handling consecutive misspellings.**Dependencies**

* **Tkinter:** The standard GUI library for Python, which facilitates the creation of windows, buttons, labels, and input fields.
* **SpellChecker:** This module is part of the pyspellchecker library, providing the core functionality for detecting spelling errors and suggesting corrections.

**Key Features**

* **Real-time Spell Checking:** Users can input a sentence, and the system checks each word against a known dictionary using SpellChecker.
* **Error Notification:** When incorrect words are detected, an error message is displayed in the application with suggestions for correction.
* **Consecutive Errors Handling**: If more than one incorrect word is detected consecutively, the system aggregates these words and displays them to the user.

**Code Overview**

* **Initialization of Spell Checker:** The SpellChecker() object is initialized to check and correct user input. It uses an internal dictionary to verify the validity of words.
* **Tracking Incorrect Words:** A global list, incorrect\_words, is used to store words that are flagged as incorrect. It is reset every time the user inputs a new sentence to prevent accumulation of errors from previous entries.
* **Checking Words Logic:** For each word in the user’s input, the function checks whether the SpellChecker suggests a different correction. If the word is incorrect, it adds it to the current\_incorrect\_words list.
* **UI Display of Errors:** If any incorrect words are found, the application updates the message\_label in red text, displaying a correction suggestion for each wrong word. If two or more incorrect words are detected, the message lists them as "Multiple incorrect words detected."
* **Handling Multiple Mistakes:** If more than one error occurs, the function aggregates and displays all the incorrect words.

**Graphical User Interface (GUI) Design**

* Main Window: The root window is created, and the title "Sentence Checker" is assigned to the application window.
* Input Field: A text input field (tk.Entry) is added to allow users to input their sentence. The field is 50 characters wide and is padded for visual spacing.
* Check Button: A button is created to allow the user to check their input. The button triggers the check\_words() function to evaluate the input sentence when clicked.
* Message Label: A label (tk.Label) is used to display messages about errors or the correctness of the sentence. The text is updated dynamically based on the input validation results.

**Application Workflow**

* The user enters a sentence into the input field.
* Upon clicking the "Check Sentence" button, the program splits the sentence into individual words.
* For each word, the SpellChecker checks whether the word exists in its dictionary or requires correction.
* If incorrect words are found, suggestions are provided, and errors are displayed in the message\_label.
* If two consecutive incorrect words are detected, a special error message is displayed listing all the incorrect words.

**Strengths**

* **Simple UI:** The application is easy to use with a clean interface, suitable for non-technical users.
* **Real-time Feedback:** Users get immediate feedback on their sentences with detailed error messages and suggested corrections.
* **Error Handling:** The application efficiently handles multiple errors and informs users about consecutive incorrect words.

**Conclusion**This spell-checking application provides a simple, effective way to check user input for spelling errors using Python's tkinter and pyspellchecker. It offers a user-friendly interface and helpful feedback on spelling mistakes. However, its functionality could be expanded to include more advanced grammar checking and sentence validation in future iterations.