SPELLING AND GRAMMAR CHECKING APP

by

SIMMY XAVIER (MST03-0069)

SUBMITTED TO META SCIFOR TECHNOLOGIES PVT LTD

****

UNDER GUIDIANCE OF

Urooj Khan

**TABLE OF CONTENTS**

1. **Abstract**
2. **Introduction**
3. **Technology Used**
4. **Dataset Information**
5. **Methodology**
6. **Code Snippet**
7. **Results and Discussion**
8. **Conclusion**
9. **References**

**ABSTRACT**

This document outlines the development of a spelling and grammar check application utilizing the autocorrect and language\_tool\_python libraries. The application performs spelling corrections on user-inputted text and identifies grammatical errors, offering suggestions for improvement. Additionally, it integrates a dictionary API to provide definitions for corrected words. The implementation involves the use of Python for text processing, error detection, and user interaction. The application demonstrates effectiveness in enhancing text clarity and correctness by providing accurate corrections and definitions.

**INTRODUCTION**

Spelling and grammar are crucial aspects of written communication, impacting clarity and professionalism. Automated tools that check and correct spelling and grammar errors are invaluable for improving text quality. This document explores the development of a spelling and grammar check application using Python libraries autocorrect and language\_tool\_python. The application leverages these tools to correct spelling mistakes, identify grammatical issues, and offer suggestions for improvements. Additionally, it integrates an online dictionary API to provide definitions for corrected words. The goal is to enhance written text by ensuring accurate spelling and grammar, and offering definitions to enrich vocabulary.

**TECHNOLOGY USED**

**TECHNOLOGY USED**

The technology stack used in this project can be summarized as follows:

**1. Programming Language**

* **Python:**
  + The application is developed in Python, a versatile language known for its simplicity and robust libraries for natural language processing.

**2. Libraries and Tools**

**1. Natural Language Processing (NLP) Libraries:**



* **Autocorrect:**
  + **Purpose:** Provides automatic spelling correction.
  + **Usage:** Utilized through the Speller class to correct individual words in the input text.
* **LanguageTool:**
  + **Purpose:** Performs grammar and style checking.
  + **Usage**: Utilized through the LanguageTool class to analyze and correct grammatical errors in sentences**.**
* **Requests:**
  + **Purpose:** Manages HTTP requests to external services.
  + **Usage:** Used to make GET requests to the DictionaryAPI to fetch word definitions.
* **DictionaryAPI (dictionaryapi.dev):**
  + **Purpose**: Provides definitions of words.
  + **Usage**: Accessed to retrieve definitions for corrected words, enhancing the user’s understanding of the corrections.

**3. Data Handling and Manipulation:**

* + **Text Input:** User-provided text is processed for spelling and grammar checking.
* **Error Detection and Correction:**
  + **Spelling Corrections:** Using autocorrect to identify and correct spelling errors.
  + **Grammar Corrections:** Using language\_tool\_python to identify and suggest fixes for grammatical errors.
* **Dictionary Integration:**
  + **Definitions:** Fetching word definitions from an online dictionary API.

**4. Web and API Integration:**

* + **requests:** A library used to interact with web APIs, particularly for fetching word definitions from an online dictionary API.
  + **pyperclip:** A Python library that allows for easy copying of text to the clipboard, used in the app's share functionality.

**5. Web Application Framework:**

* + **Streamlit:** A Python library used to create the interactive web application. It allows users to input text, check for spelling and grammar errors, and receive corrected text in real-time.

**6. Web and App Development Tools:**

* + **Markdown:** Used within Streamlit to style and format the corrected text output in the web application.
  + **Download and Share Features:** Implemented using Streamlit’s download and button functionalities, allowing users to share or download corrected text.

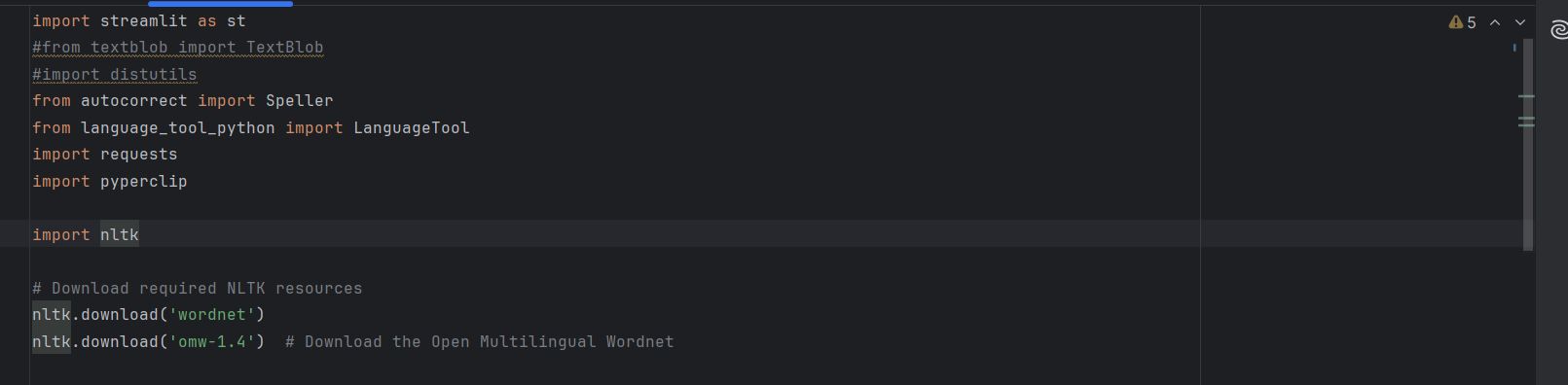
These technologies collectively support the creation of a real-time, user-friendly spelling and grammar check application.

**DATASET INFORMATION**

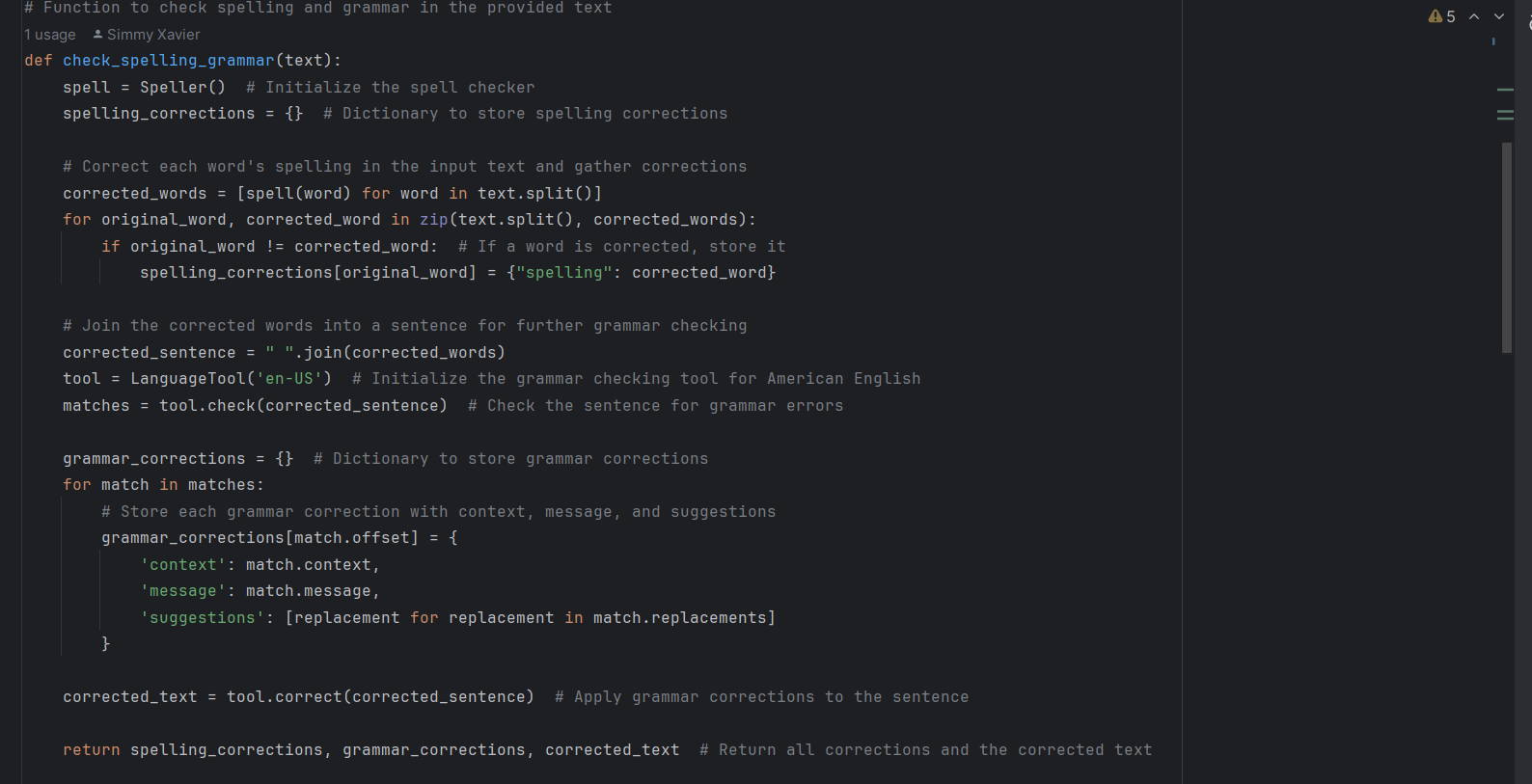
The application does not use a specific dataset but operates on user-provided text input. It performs spelling and grammar checks, and fetches definitions from an online dictionary API as needed.

**METHODOLOGY**

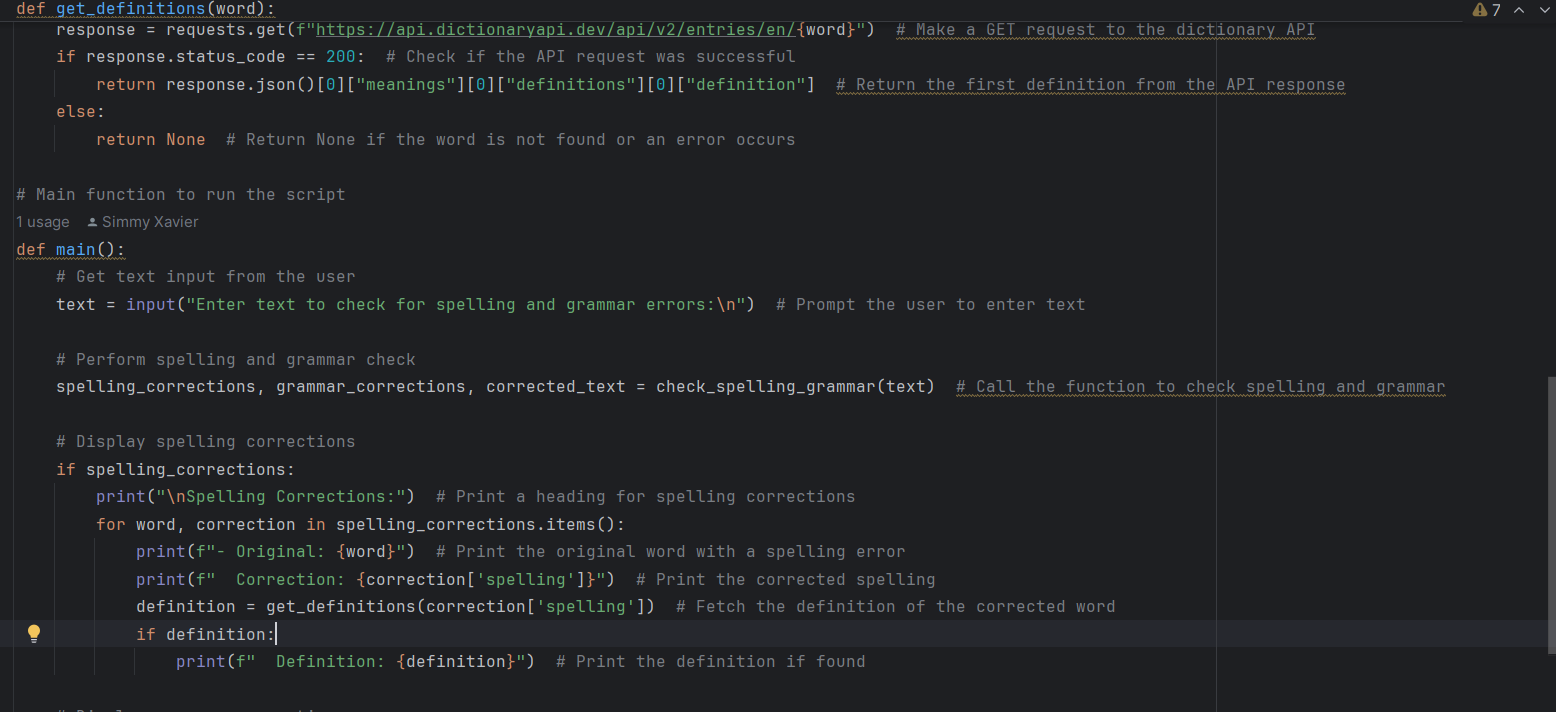
* The methodology employed in this project includes the following steps:
* **Importing Libraries:**
  + Libraries such as autocorrect, language\_tool\_python, and requests are imported for text processing, error checking, and dictionary access.



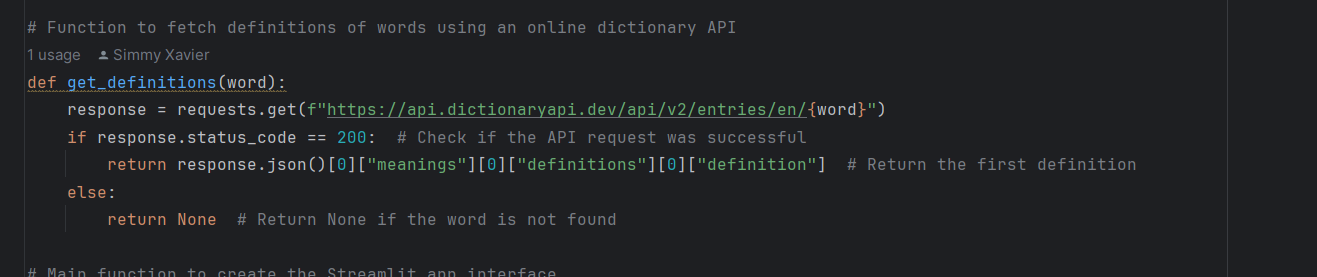
* **Spelling and Grammar Checking:**
  + The check\_spelling\_grammar function:
    - Uses autocorrect to correct spelling errors and gather corrections.
    - Applies language\_tool\_python to check for grammatical errors, providing corrections and suggestions.
    - Returns spelling and grammar corrections, along with the corrected text.



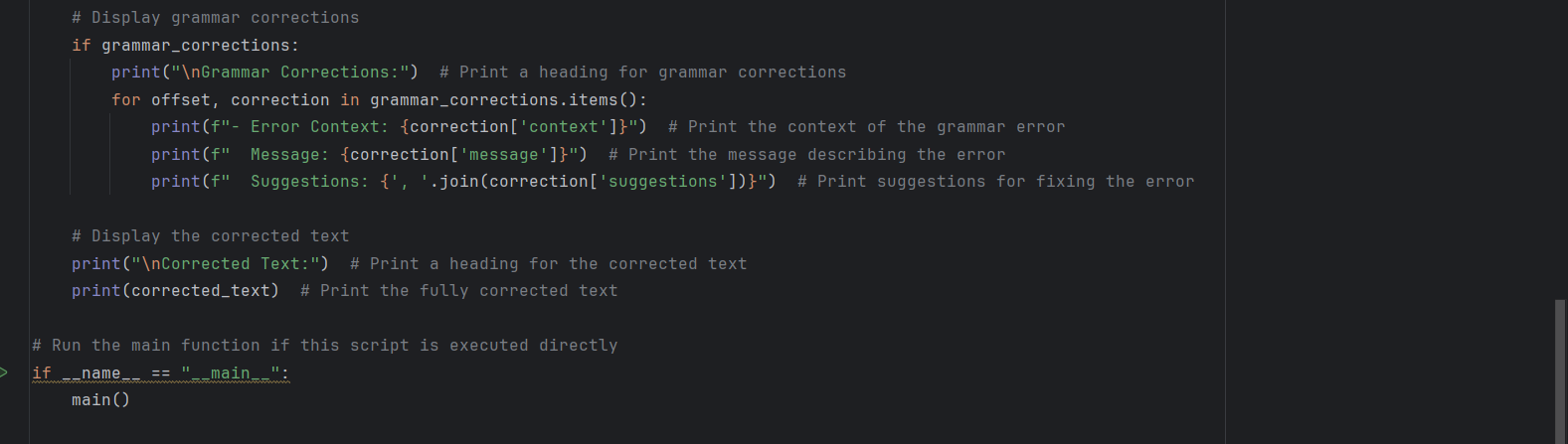
* **Spelling Correction**
* Initialization:
  + The Speller class from the autocorrect library is initialized to handle spelling correction.
* Correction Process:
  + The input text is split into words.
  + Each word is checked and corrected if necessary using the Speller class.
  + Corrections are stored in a dictionary, mapping original words to their corrected versions.
* Sentence Reconstruction:
  + The corrected words are joined into a complete sentence for further grammar checking.
* **Grammar Checking**
* Analysis:
  + The reconstructed sentence is passed to LanguageTool for grammar and style analysis.
* Error Detection:
  + The tool identifies grammatical errors, providing detailed feedback including error context, message, and suggested corrections.
* **Application of Corrections:**
  + Suggested corrections are applied to produce a fully corrected version of the text.
* **Word Definition Retrieval**
* **API Request:**
  + For each corrected word, a GET request is made to the DictionaryAPI to obtain its definition.



* **Response Handling:**
  + If the API request is successful, the definition is extracted and presented.
    - * If the word is not found, no definition is provided.

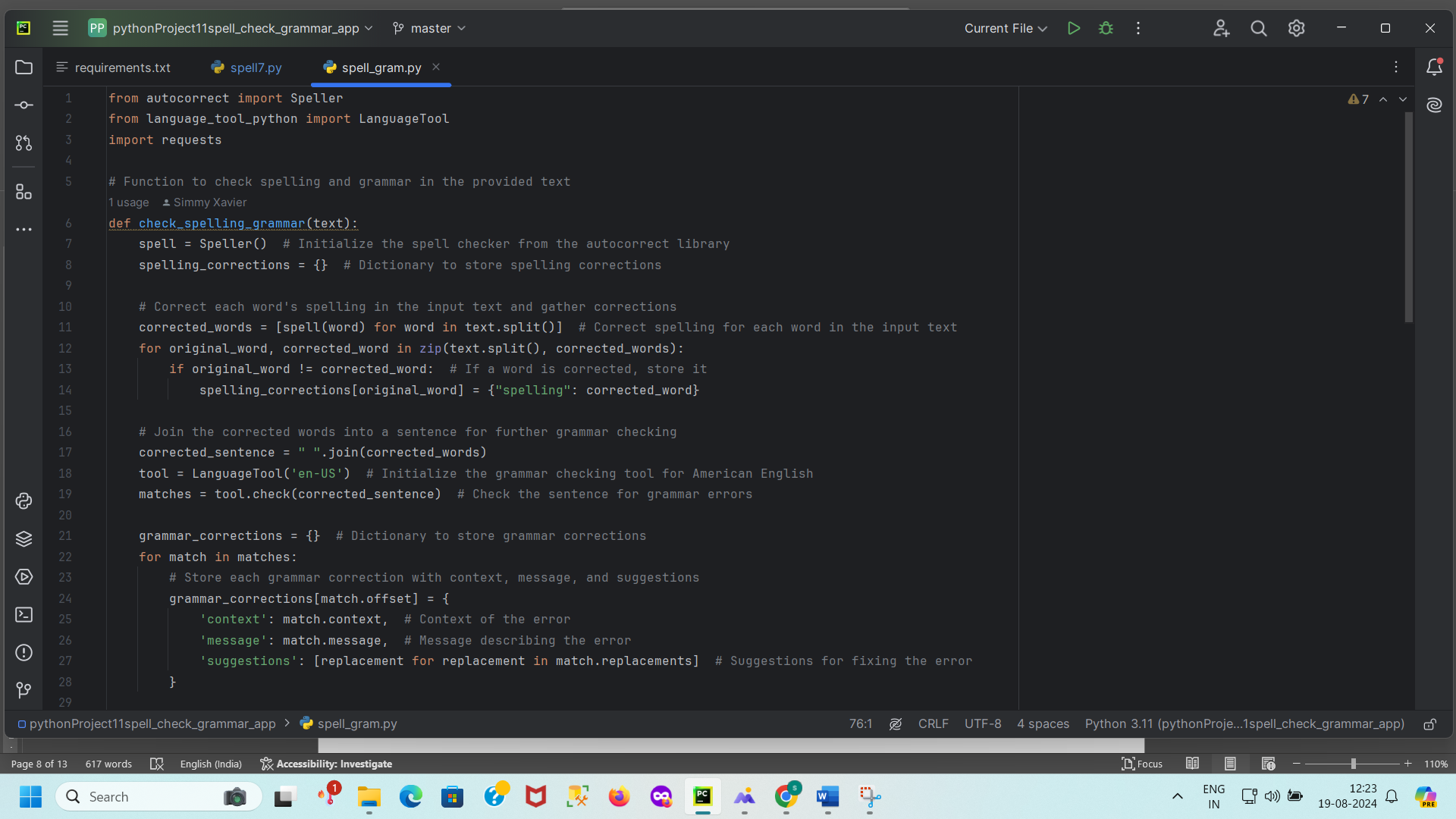


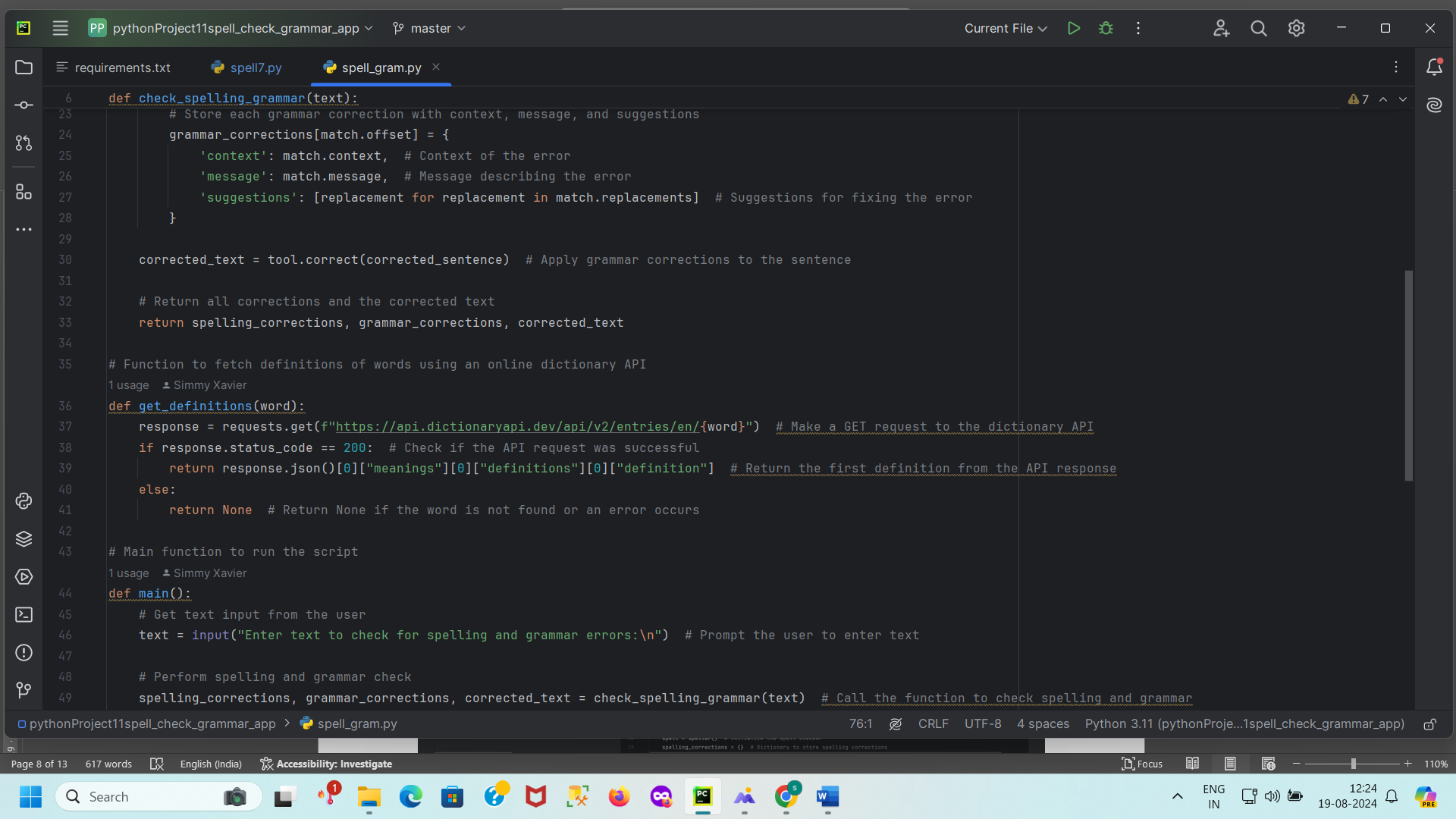
* **Fetching Definitions:**
  + The get\_definitions function:
    - Retrieves definitions for corrected words from an online dictionary API.
    - Handles API responses and provides definitions or indicates if a word is not found.
* **User Interaction:**
  + Text Input: The main function prompts the user to enter text.
  + Displaying Results: Outputs spelling corrections, grammar suggestions, and corrected text to the user.

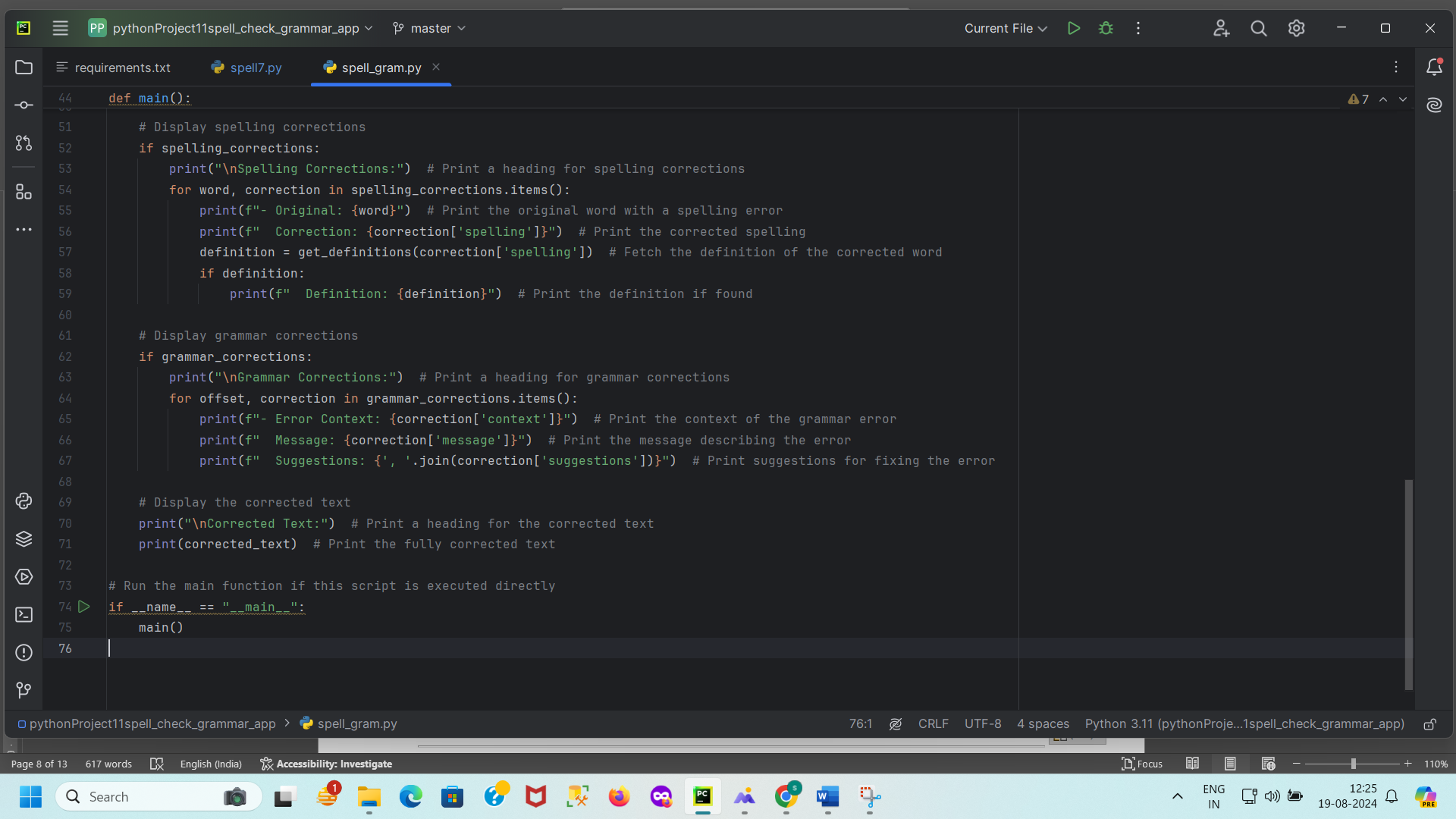


* + Exception Handling: Manages errors during processing and provides feedback.
    - User Interaction (in a Streamlit App Context)
* **Text Input:**
  + Users provide text via a text area in the Streamlit app.
* **Processing:**
  + On submission, the text undergoes spelling and grammar checks as described.
* **Feedback:**
  + Spelling and grammar corrections are displayed, along with definitions of corrected words.
  + The fully corrected text is shown to the user, offering a polished version of their input.

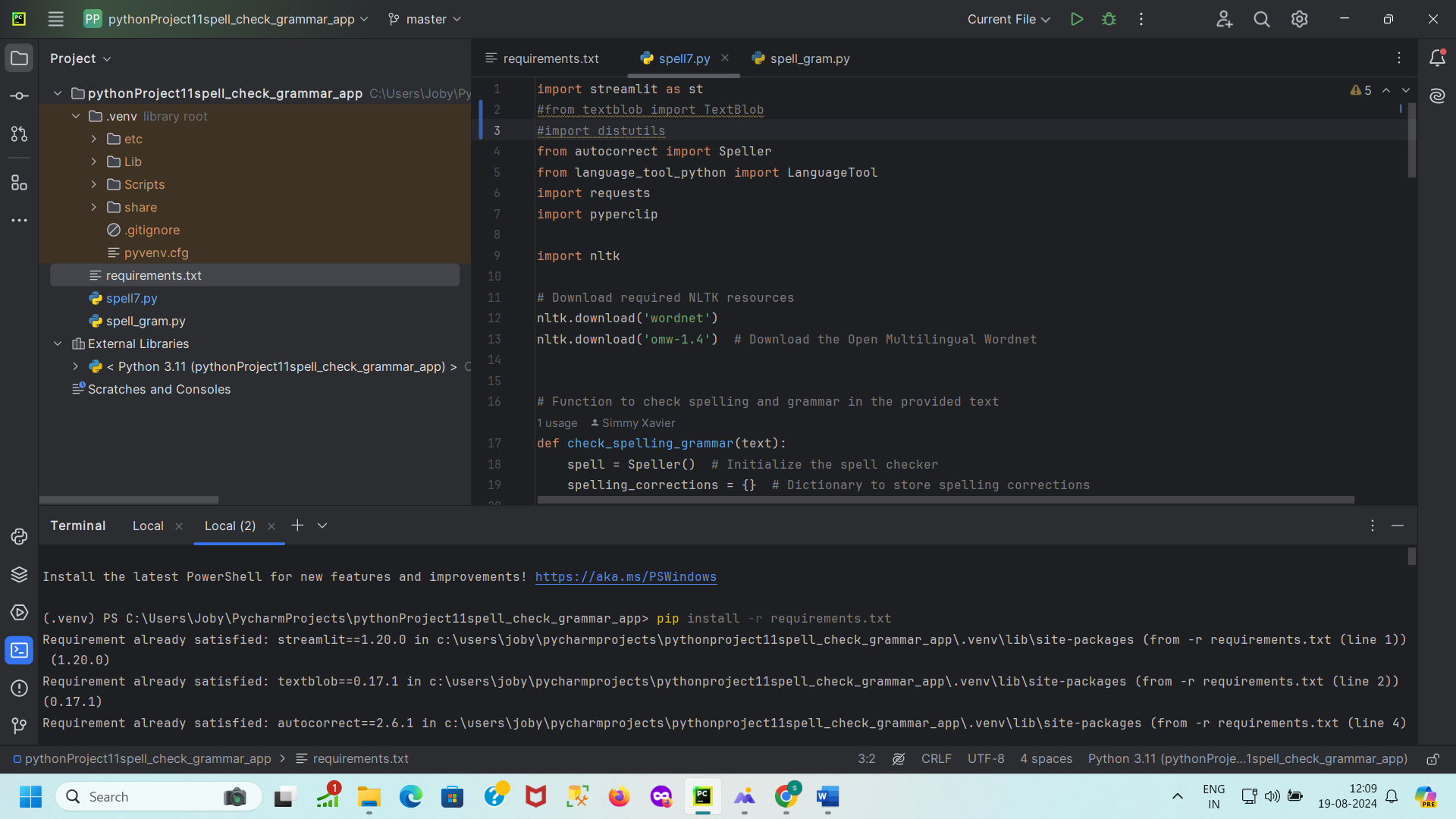
**CODE SNIPPET**

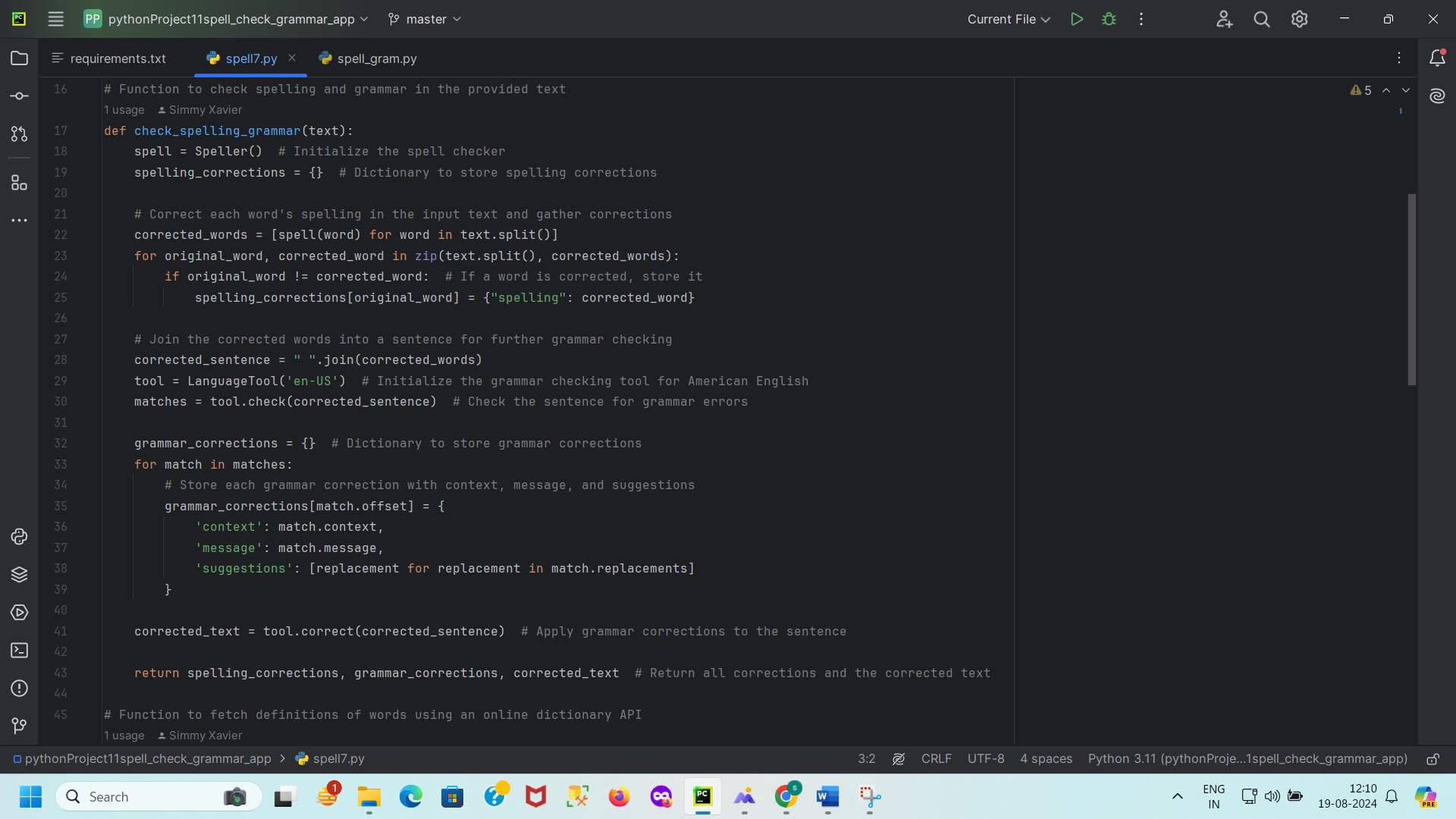


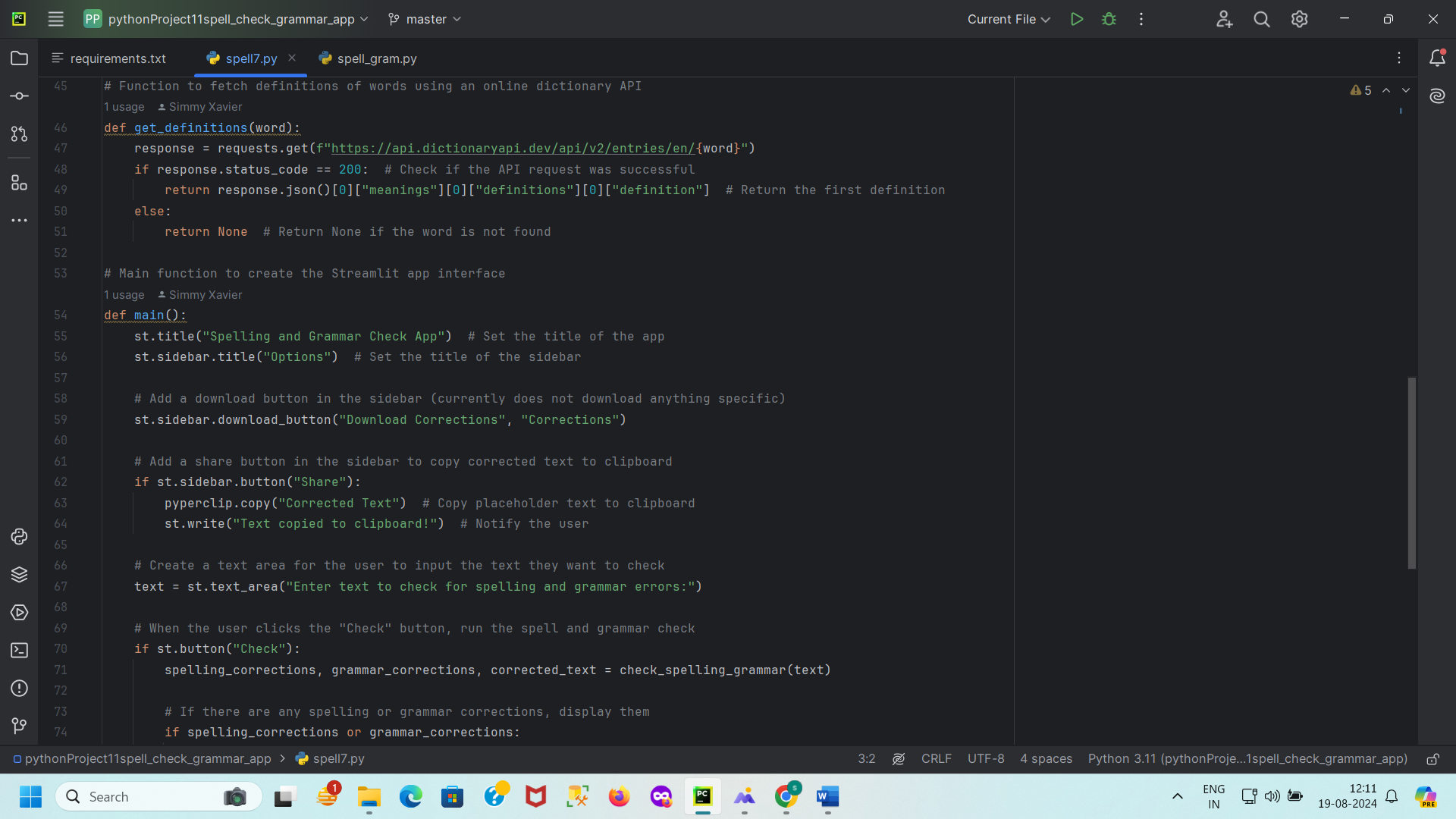


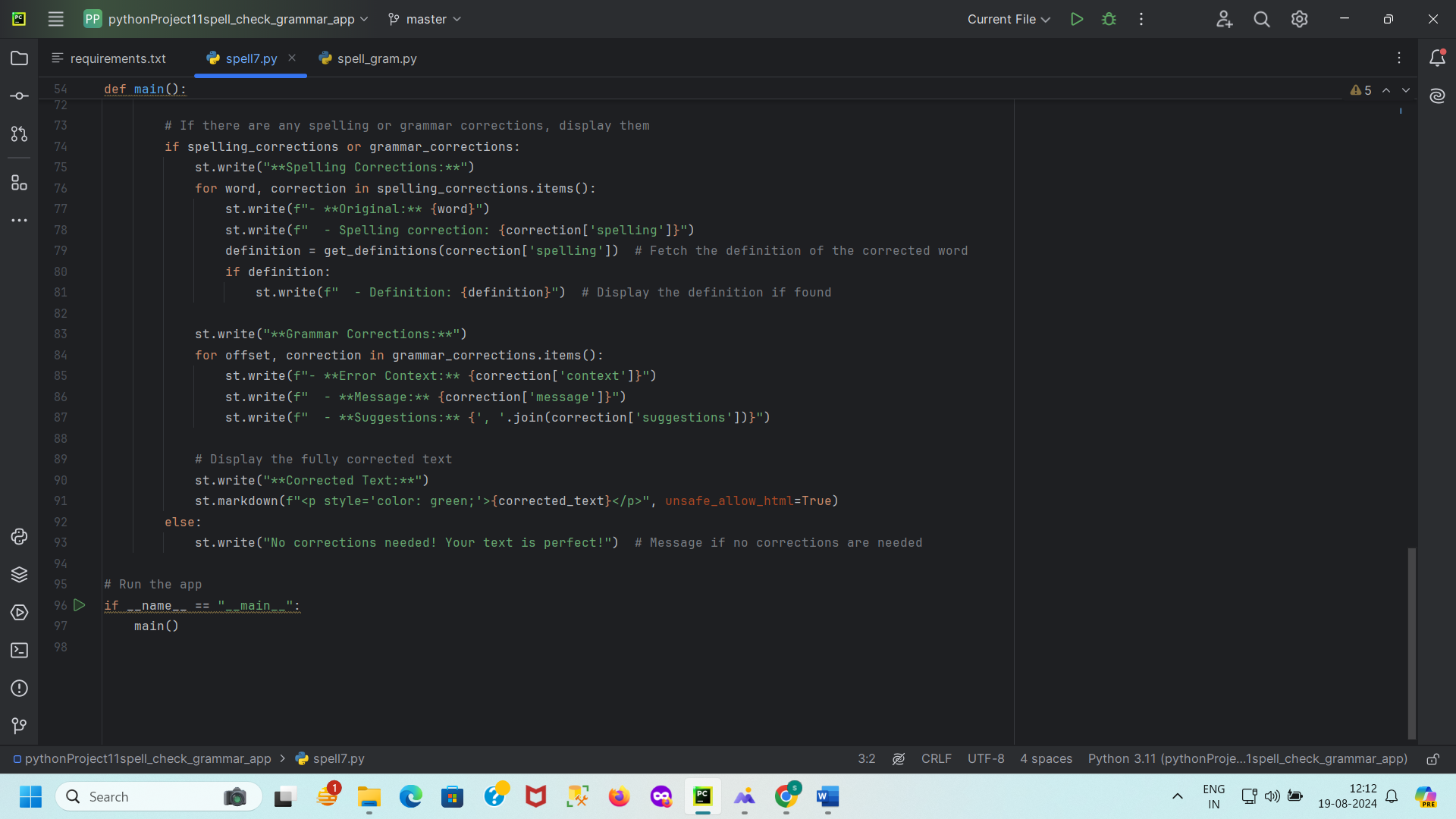


**STREAMLIT CODE**

****

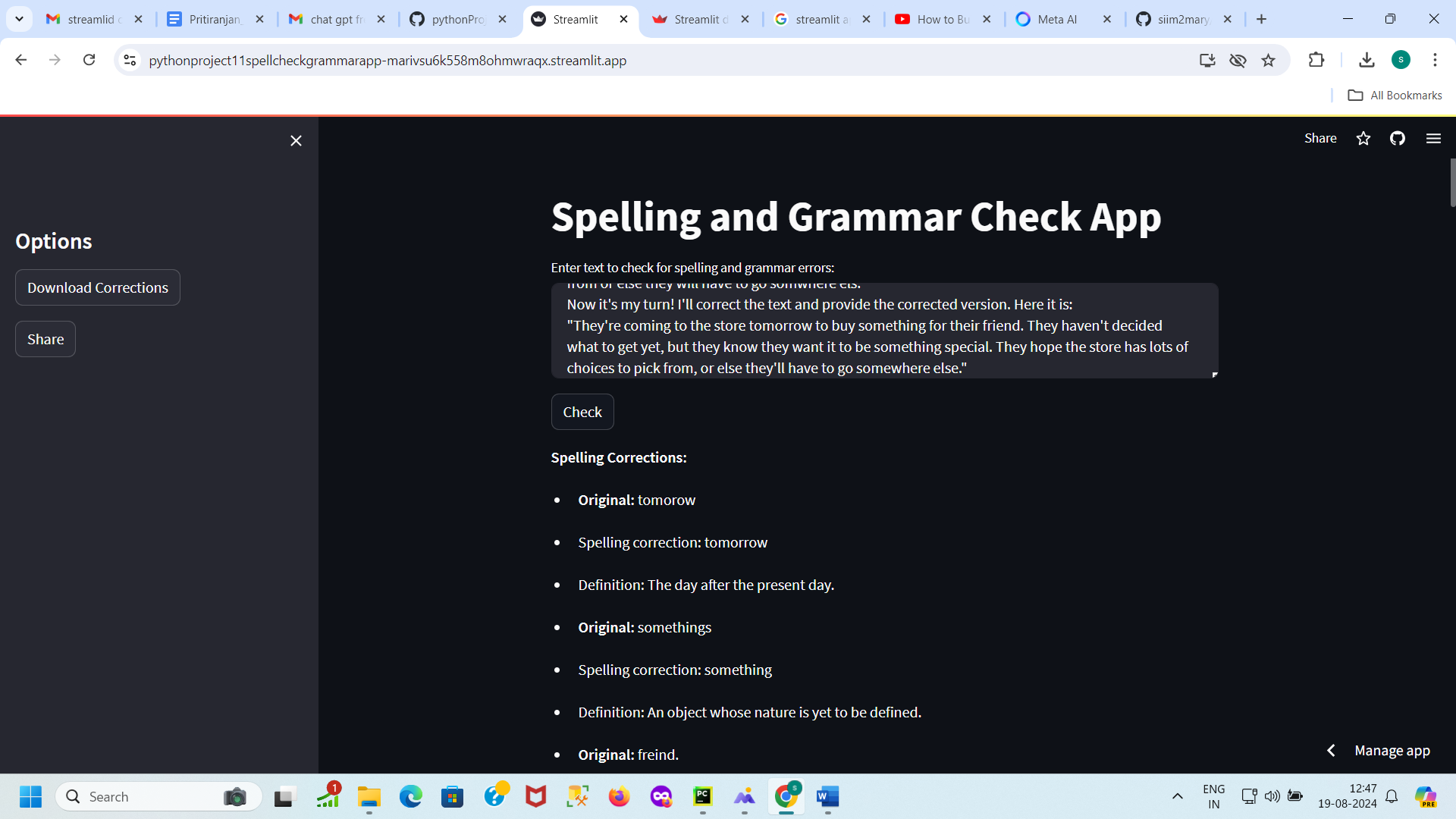
****

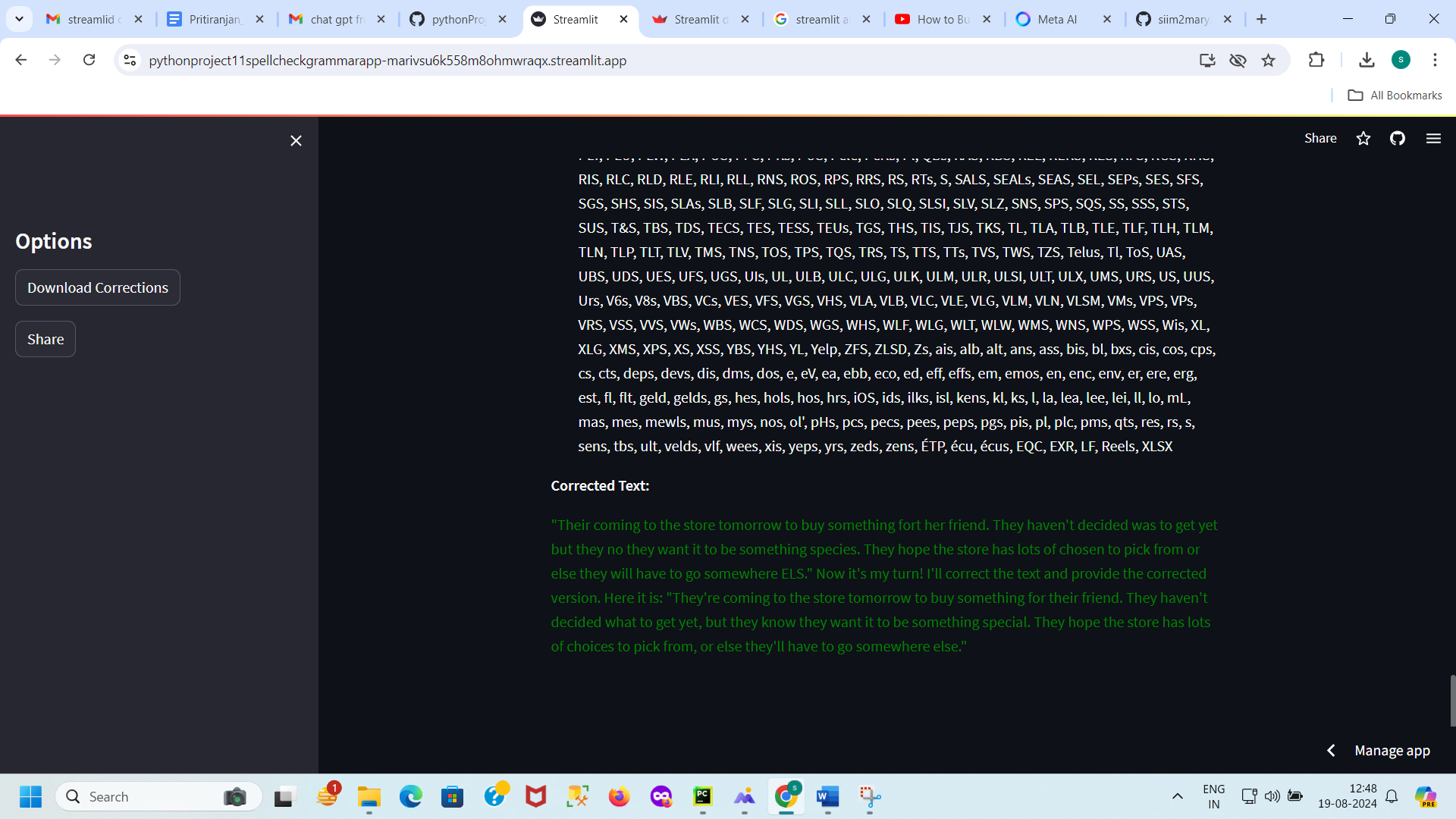
****

****

**RESULTS AND DISCUSSION**

The spelling and grammar check application effectively corrects spelling mistakes and identifies grammatical errors. The spelling corrections are accurate, with the application providing alternative suggestions and definitions. Grammar corrections are appropriately identified, with clear suggestions for improvement. The application enhances text clarity and correctness, demonstrating its utility in improving written communication.



Streamlit cloud deployed app link: https://pythonproject11spellcheckgrammarapp-marivsu6k558m8ohmwraqx.streamlit.app/

The Spelling and Grammar Check Application has been developed with the following key results:

1. **Spelling Corrections:**
   * The application effectively identifies and corrects spelling errors in the input text. Each misspelled word is detected and replaced with its corrected form. Additionally, definitions of corrected words are fetched from an online dictionary API, enhancing user understanding of the corrections.
2. **Grammar Corrections:**
   * The application utilizes LanguageTool to detect and correct grammatical errors. It provides detailed information on each grammar issue, including the context, message describing the error, and suggestions for corrections. This helps users understand and improve their grammatical accuracy.
3. **User Interface:**
   * The application features an intuitive and interactive web interface created with Streamlit. Users can input text, view spelling and grammar corrections, and receive the corrected text directly in the web app. The interface also includes functionalities for copying the corrected text to the clipboard and a placeholder for downloading corrections.
4. **Text Preprocessing:**
   * Input text is preprocessed to handle spelling corrections before grammar checking. The application processes each word to correct spelling, then assembles the corrected words into a coherent sentence for grammar analysis.
5. **Definitions and Suggestions:**
   * Definitions of corrected words are provided to offer additional context and understanding. Grammar issues are highlighted with suggestions for improvement, enhancing the user's ability to refine their writing.

**CONCLUSION**

This project successfully implements a spelling and grammar check application using Python. The application provides accurate spelling corrections, grammatical error identification, and word definitions, enhancing text quality. Future improvements could include expanding the dictionary API integration and optimizing performance for larger texts.

The Spelling and Grammar Check Application demonstrates a robust solution for enhancing written communication by leveraging advanced text-processing tools:

* **Effectiveness:** The integration of the autocorrect library and LanguageTool provides comprehensive spelling and grammar checking. The application successfully corrects spelling errors and offers detailed grammar corrections, improving text quality.
* **User Experience:** The Streamlit-based web interface ensures a user-friendly experience. Users can interact with the application seamlessly, check their text, and receive instant feedback on corrections.
* **Utility:** By combining spelling correction with grammar checking and definition fetching, the application serves as a valuable tool for anyone looking to improve their writing skills. It supports users in creating error-free, grammatically accurate text while also providing educational value through definitions.
* **Future Enhancements:** Future improvements could include:
  + Expanding language support for grammar checking.
  + Enhancing the dictionary API integration for more comprehensive word definitions.
  + Adding additional text analysis features, such as style and tone checking.

Overall, the application meets its objective of providing an effective spelling and grammar checking solution, contributing to better writing and communication.

**REFERENCES**

1. **Speller Documentation:** <https://pypi.org/project/autocorrect/>
2. **LanguageTool Documentation:** <https://languagetool.org/>
3. **Dictionary API Documentation:** <https://dictionaryapi.dev/>
4. **Python Programming Language:**
   1. Python Software Foundation. (n.d.). Python. Retrieved from <https://www.python.org/>
5. **Autocorrect Library:**
   1. autocorrect. (n.d.). GitHub. Retrieved from <https://github.com/fsondej/autocorrect>
6. **LanguageTool:**
   1. LanguageTool. (n.d.). LanguageTool - Online grammar and style checking. Retrieved from <https://languagetool.org/>
7. **Requests Library:**
   1. Kenneth Reitz. (2017). Requests: HTTP for Humans. Retrieved from https://docs.python-requests.org/
8. **DictionaryAPI:**
   1. DictionaryAPI. (n.d.). Dictionary API. Retrieved from <https://dictionaryapi.dev/>
9. **Streamlit:**
   1. Streamlit. (n.d.). Streamlit: The fastest way to build data apps. Retrieved from <https://streamlit.io/>
10. **NLTK (Natural Language Toolkit):**
    1. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media. Retrieved from <https://www.nltk.org/>
11. **NLTK WordNet:**
    1. WordNet. (n.d.). WordNet. Retrieved from <https://wordnet.princeton.edu/>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*