**Experiment 10**

**Aim: Mini project report and implementation on any 1 real world application of NLP**

**Theory:**

**Abstract**

In this project, we aim to develop a grammar autocorrection system using Natural Language Processing (NLP) techniques. The system leverages two powerful Python libraries: Natural Language Toolkit (NLTK) and Pattern. The primary goal is to build a tool that detects and corrects grammatical errors in text, enhancing the accuracy and readability of written content.

The NLTK library is employed for text processing tasks such as tokenization, part-of-speech tagging, and parsing. It enables the system to break down sentences into grammatical components and identify errors based on predefined syntactic rules. Additionally, Pattern is used to further enhance the system by providing advanced grammatical analysis and correction capabilities. Pattern’s modules offer functions for identifying verb tenses, moods, and syntactic relationships, which are crucial in pinpointing grammatical inconsistencies.

Our system works by analyzing user input, identifying grammatical errors such as incorrect verb conjugations, subject-verb agreement issues, and incorrect word order, and suggesting corrections in real-time. The combination of NLTK's robust text processing capabilities with Pattern's grammar analysis provides an efficient framework for building a comprehensive grammar correction tool. This project showcases how linguistic and statistical approaches in NLP can be integrated to assist in improving written communication.

Key applications of this system include integration into writing assistants, educational tools for language learning, and automated text editing software.

**Introduction**

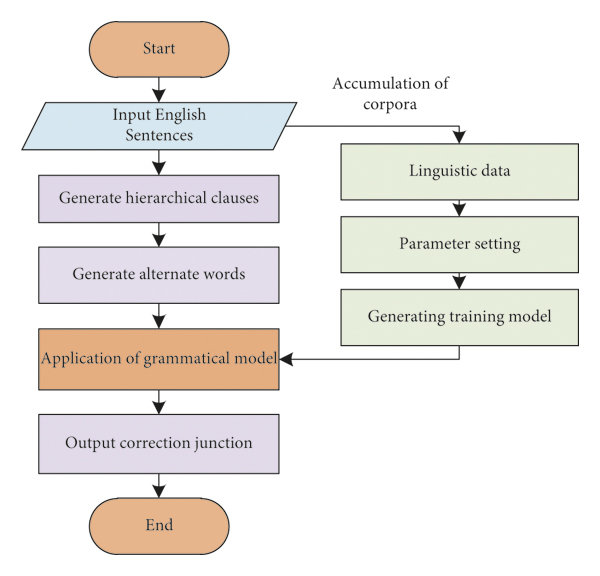
In the digital age, effective written communication is essential in many domains, from academic writing and business correspondence to social media interactions. However, ensuring grammatical correctness can be challenging, particularly for non-native speakers and individuals with limited proficiency in formal language rules. This has led to an increasing demand for tools that can automatically detect and correct grammatical errors in real time. The Grammar Autocorrect NLP Project addresses this need by utilizing Natural Language Processing (NLP) techniques to build an intelligent system that identifies and corrects grammatical mistakes in text.

The project leverages two widely used NLP libraries: Natural Language Toolkit (NLTK) and Pattern. NLTK provides powerful tools for basic text processing tasks, such as tokenization, part-of-speech tagging, and syntactic parsing. These functionalities are essential for breaking down sentences into their grammatical components and detecting anomalies. On the other hand, Pattern offers advanced linguistic features, including verb conjugation, tense identification, and syntactic relationships, which allow for deeper grammatical analysis and correction.

The combination of these two libraries allows the system to perform tasks such as detecting subject-verb agreement errors, incorrect verb tenses, and other grammatical inconsistencies. Once detected, the system provides suggestions to correct these errors, improving both the accuracy and fluency of the text.

This project is aimed at developing a reliable and scalable grammar correction tool that can be used in various applications, such as writing assistants, language learning platforms, and automated text editing systems. By automating grammar correction, we aim to assist users in producing grammatically sound content with minimal effort, contributing to more effective and polished written communication.

**System Design**



**Working of the Project**

1. User Input

Action: The process begins when a user inputs text that needs to be grammatically checked and corrected.

Output: Raw text is captured for processing.

2. Preprocessing Module

Purpose: Prepare the input text for further analysis by cleaning and normalizing it.

Steps:

Tokenization:

Action: The text is split into individual tokens (words and punctuation marks) to make it easier to analyze.

Output: A list of tokens (e.g., ["I", "am", "going", "to", "the", "store."]).

Text Normalization:

Action: Convert the tokens to a standard format, which may include lowercasing all words, removing unnecessary punctuation, or correcting common typos.

Output: A clean version of the input text (e.g., "i am going to the store").

3. NLTK Module

Purpose: Analyze the grammatical structure of the text.

Steps:

Part-of-Speech (POS) Tagging:

Action: Each token is tagged with its grammatical role (e.g., noun, verb, adjective).

Output: A list of tuples with tokens and their POS tags (e.g., [("i", "PRP"), ("am", "VBP"), ("going", "VBG"), ...]).

Syntactic Parsing:

Action: Analyze the sentence structure to identify phrases and grammatical relationships.

Output: A parse tree that represents the grammatical structure of the sentence.

4. Pattern Module

Purpose: Perform deeper grammatical analysis and error detection.

Steps:

Grammatical Error Detection:

Action: The system checks the input against grammatical rules to identify errors such as:

Subject-verb agreement issues (e.g., "She go to the store" vs. "She goes to the store").

Incorrect verb tenses (e.g., "I seen" instead of "I saw").

Output: A list of identified errors and their locations in the text.

Error Correction Suggestion:

Action: For each detected error, the system generates suggestions for correction based on grammatical rules.

Output: Suggested corrections for each identified error.

5. Correction Engine

Purpose: Integrate the suggestions and produce the corrected output.

Steps:

Compare Original & Suggested Corrections:

Action: The engine compares the original text with the suggested corrections, evaluating which suggestions to apply.

Output: A new version of the text with corrections highlighted or annotated.

Highlight/Auto-correct Errors:

Action: The system either highlights the errors for the user to review or automatically applies the corrections.

Output: The final corrected text.

6. Output Module

Purpose: Present the results to the user.

Steps:

Display Corrected Text:

Action: The system displays the corrected text to the user.

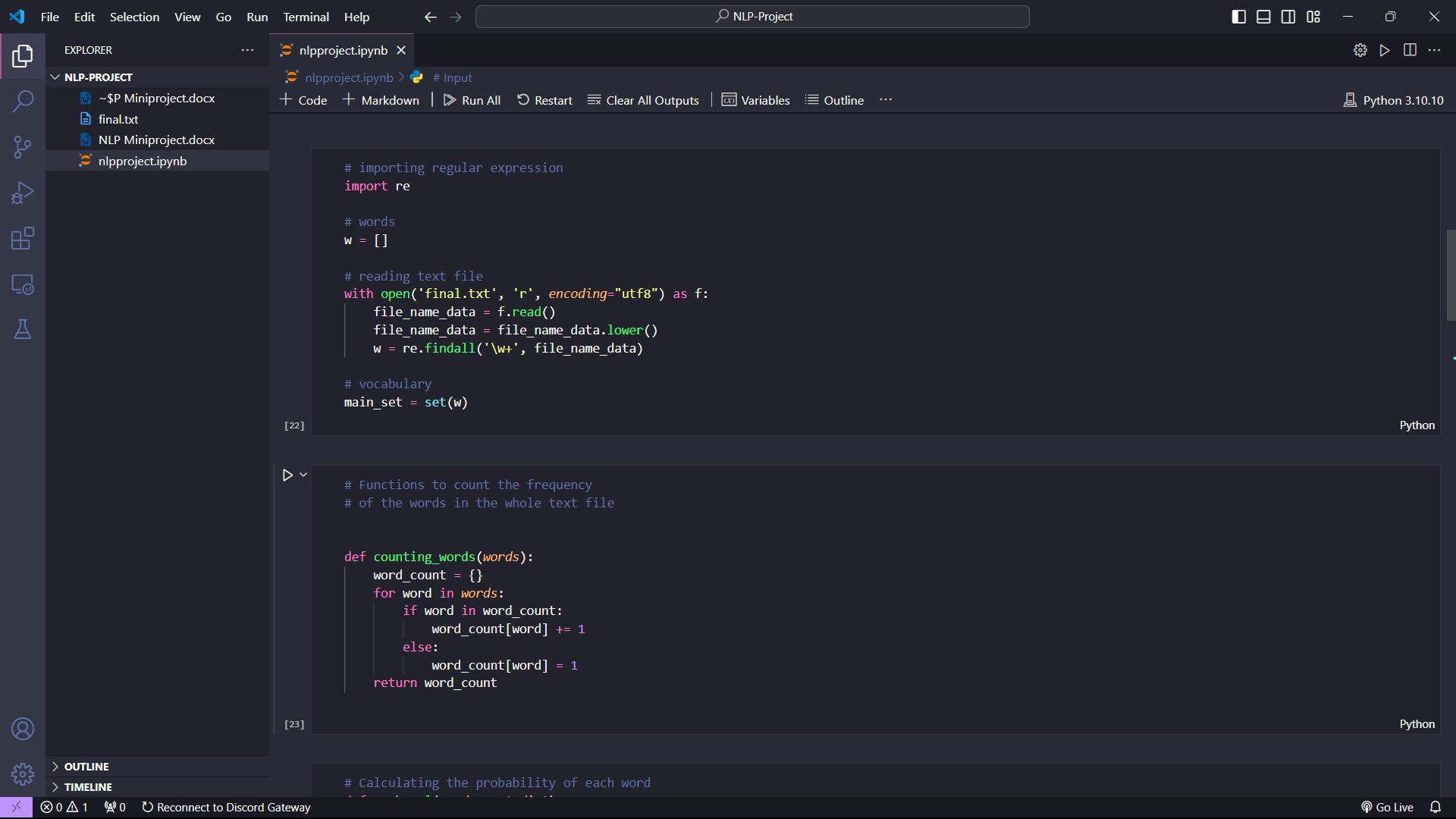
Output: A visually clear representation of the corrected text, possibly with errors highlighted.

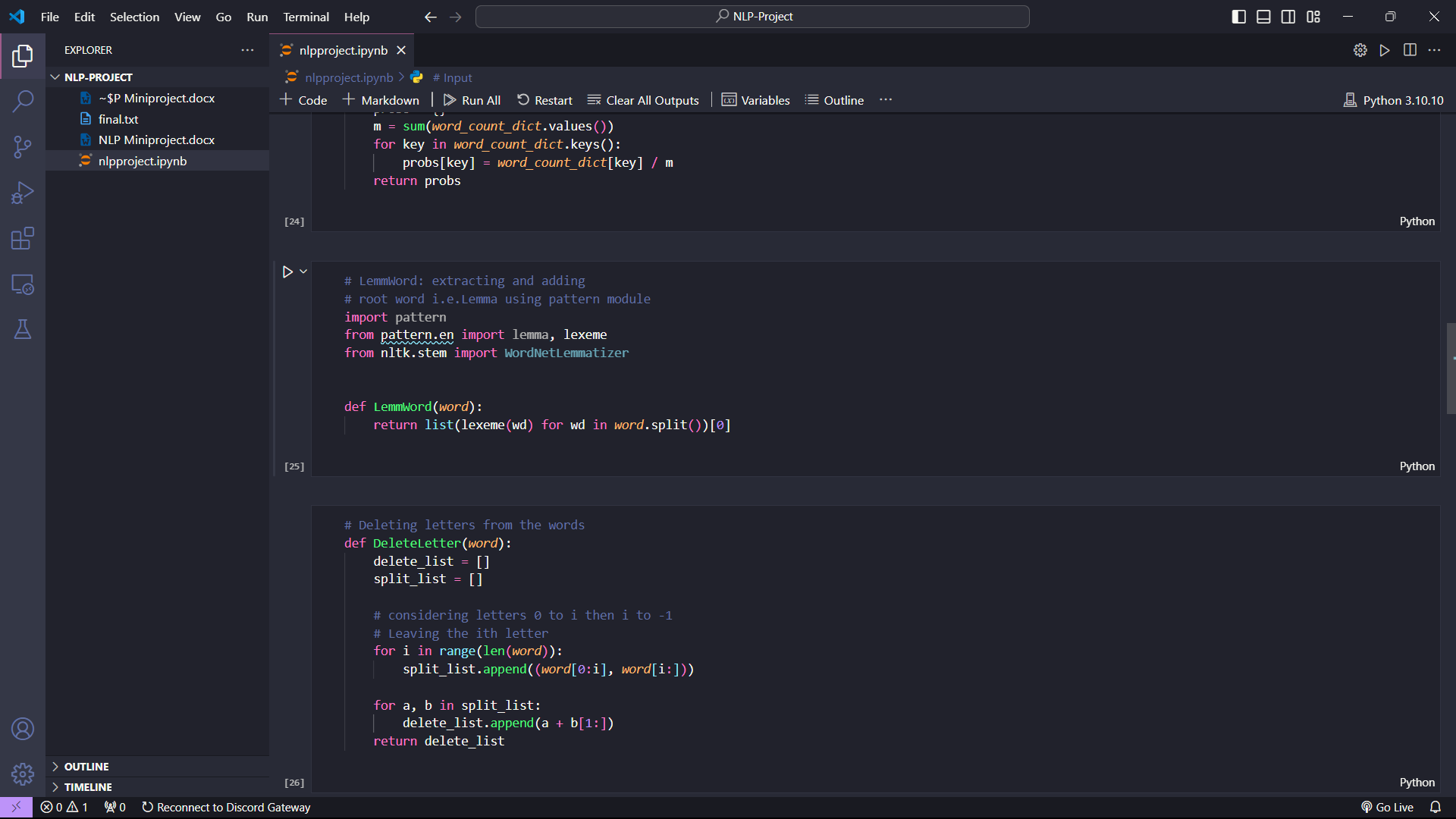
Provide Suggestions/Explanations:

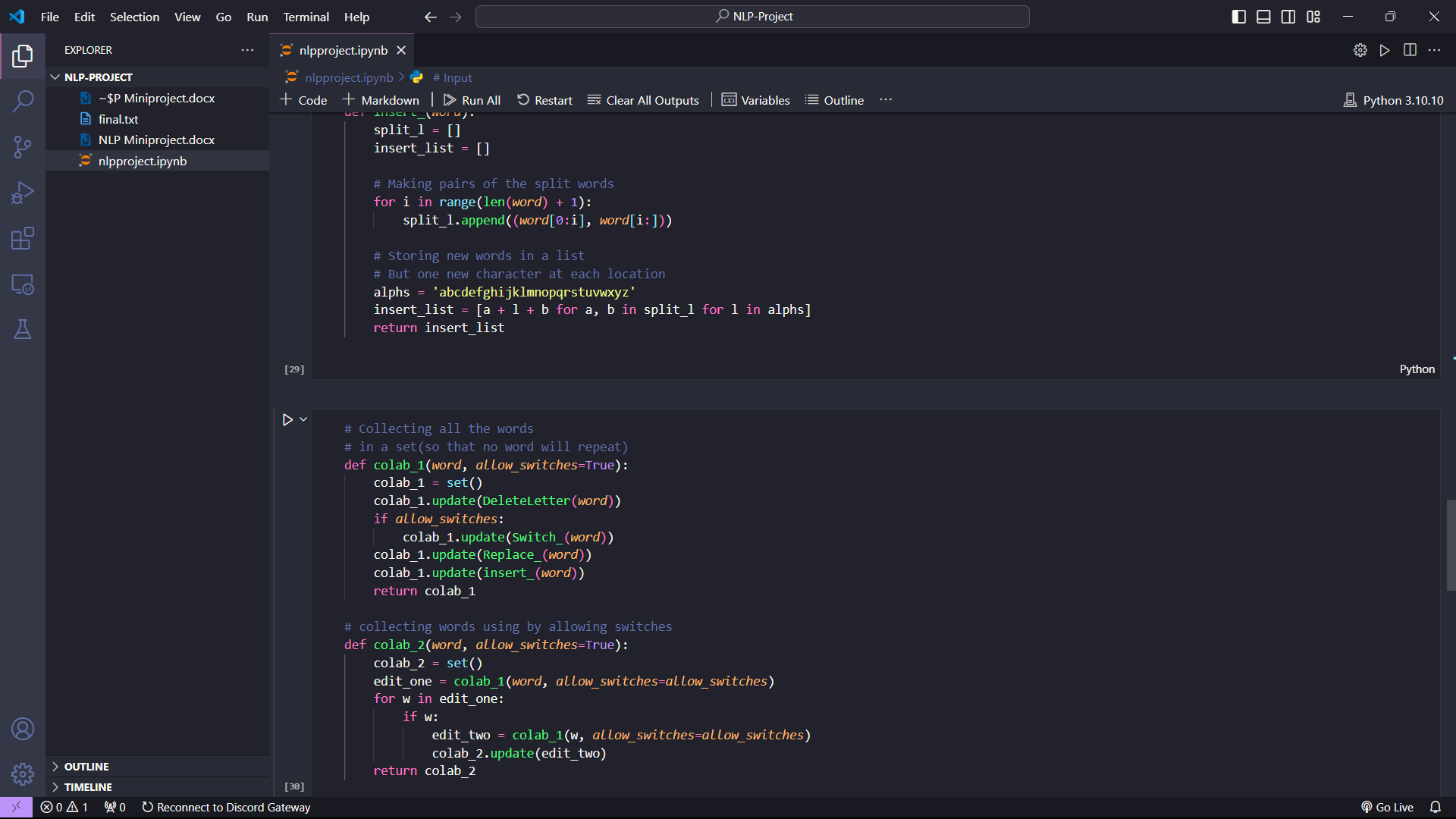
Action: Along with the corrected text, the system may offer explanations for the changes made or additional suggestions for improvement.

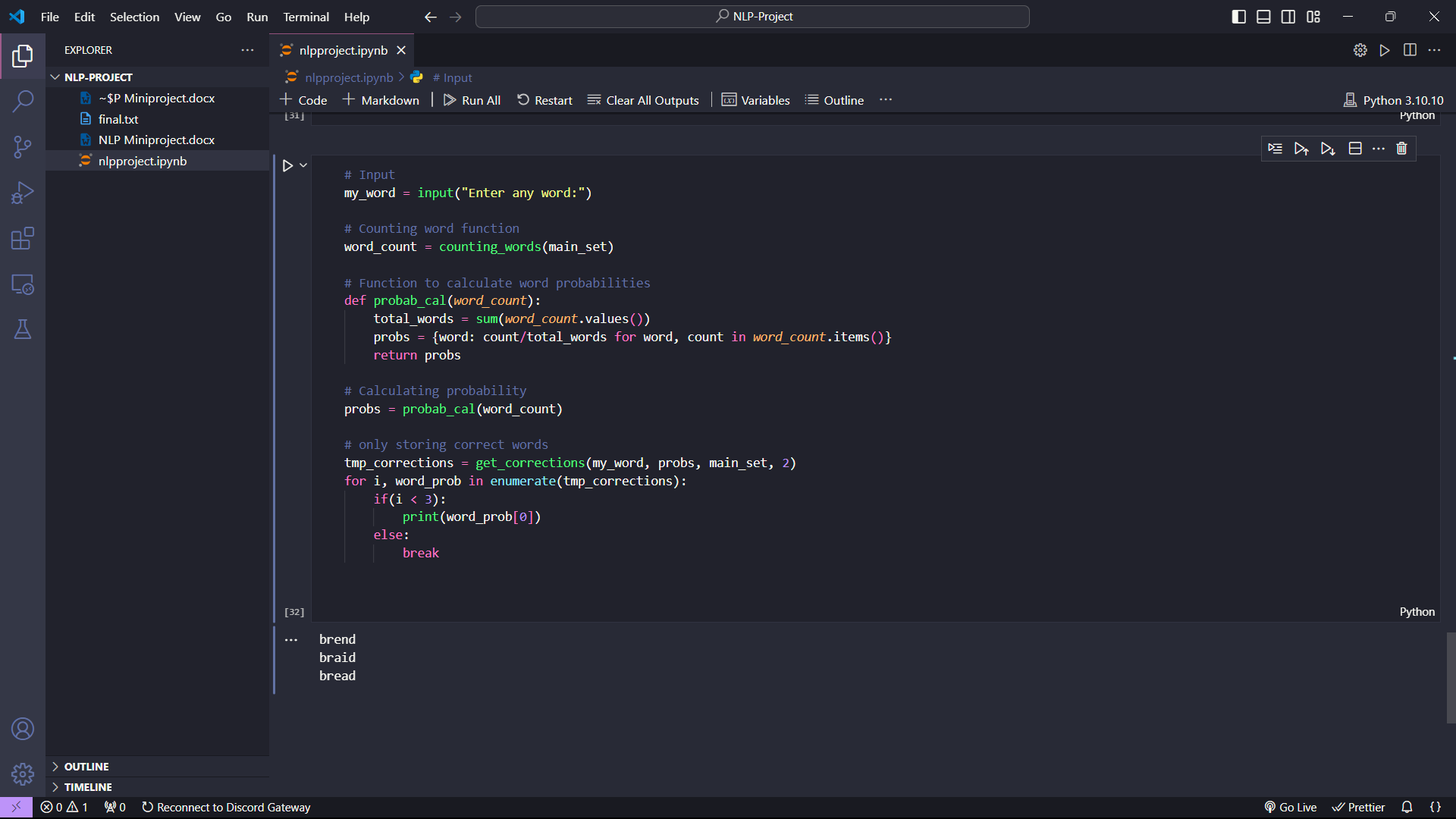
Output: An interactive interface where users can see their original text, the corrections, and the rationale behind them.

**Screenshots**









**Conclusion**: Hence we made a mini project report and implementation on Grammar Autocorrect.