Plagiarism Checker Report

Objective:
The Plagiarism Checker project is designed to offer a simple yet effective tool for comparing the similarity between two text files. Its primary objective is to aid in the detection of potential plagiarism by analyzing how closely two documents match. The project features a user-friendly Graphical User Interface (GUI) built with Tkinter, ensuring ease of use. Users can select two text files using file selection buttons within the interface. Once files are selected, they have the option to view the contents of each file separately.Upon initiating the comparison process, the tool cleans both texts by converting them to lowercase, removing non-alphanumeric characters, and eliminating extra spaces to ensure accurate comparison. The similarity between the two files is then calculated and displayed as a percentage, using functionalities from the difflib library. This project aims to provide educators, researchers, and writers with a straightforward means to assess textual similarity, thereby supporting efforts to maintain academic integrity and originality.

Purpose:
The purpose of this project is to provide users with an easy-to-use application that leverages text comparison algorithms to detect similarities in documents. By identifying textual overlap, the tool assists in maintaining the integrity and originality of written content.

Features:
- Graphical User Interface (GUI) created using Tkinter, making it user-friendly.
- Allows users to select and view two text files for comparison.
- Displays the similarity percentage between the files.
- Cleans the input text by converting it to lowercase, removing non-alphanumeric characters, and eliminating extra spaces.
- Uses the difflib library to compare the cleaned texts and calculate the similarity ratio.

Key Components:
- *GUI Elements:* Entry widgets for file paths, buttons to select and view files, a button to trigger the comparison, and a label to display the similarity result.
- *Text Processing:* Cleans and prepares the text for comparison, ensuring that superficial differences do not affect the similarity measurement.
- *Similarity Algorithm:* Uses the SequenceMatcher class from the difflib library to calculate the similarity ratio between the two cleaned text files.

Instructions:
1. Run the script to launch the GUI.
2. Click "Select File 1" to choose the first text file.
3. Click "Select File 2" to choose the second text file.
4. Optionally, click "View File 1" and "View File 2" to view the contents of the selected files.
5. Click "Compare Files" to compare the similarity between the two files.
6. The similarity percentage will be displayed in the result label.

Key Functions:
1. *clean_text(text):* Cleans the input text by converting to lowercase, removing non-alphanumeric characters, and extra spaces. This ensures that the comparison is not affected by case sensitivity or special characters.

2. *open_file1():* Opens a file dialog to select the first text file and updates the corresponding entry widget with the selected file path.

3. *open_file2():* Opens a file dialog to select the second text file and updates the corresponding entry widget with the selected file path.

4. *view_file_content(file_path):* Opens a new window displaying the contents of the selected file, allowing the user to review the file's content within the application.

5. *compare_files():* Reads the contents of the two selected files, cleans the text, and uses SequenceMatcher from the difflib library to calculate the similarity ratio. The similarity percentage is then displayed in the result label.

Summary:
The Plagiarism Checker project offers a practical and efficient way to compare the contents of two text files, making it an invaluable tool for detecting plagiarism. Its user-friendly interface and robust text processing capabilities ensure accurate and meaningful comparisons, helping users uphold the originality of their work. Whether in academic research, content creation, or professional documentation, this tool provides a reliable solution for maintaining content integrity.