

Project: TITLE-WORD FREQUENCY COUNTER
by Akula Charith
Domain: AI, ML and Data Science

PROJECT DESCRIPTION:

A Python console application that reads a text file, analyzes its content, and shows how many times each unique word appears. The program should handle input/output, cleaning up punctuation/case, sorting, and display summary statistics.

Source Code:

```
import os
import
string
from collections import Counter

def clean_text(text):
    """Convert text to lowercase, remove punctuation, and split into words."""
    # Convert to lowercase
    text = text.lower()
    # Remove punctuation
    translator = str.maketrans("", "", string.punctuation)
    text = text.translate(translator)
    # Split into words
    words = text.split()
    return words

def get_top_n(counter, n):
    return counter.most_common(n)

def word_frequency_counter():
    print("=== Word Frequency Counter")
    print("===")
    while True:
        file_path = input("Enter the path to a text file: ").strip()
        if not os.path.isfile(file_path):
            print("File not found. Please try again or type 'exit' to quit.")
            if file_path.lower() == "exit":
                return
            continue
        else:
            break

    try:
        with open(file_path, "r", encoding="utf-8") as file:
            text = file.read()
    except Exception as e:
        print(f"Error reading file: {e}")
    return

words = clean_text(text)
```

```

if not words:
    print("The file is empty. No words to analyze.")
    return

word_counts = Counter(words)

# Step 4: Display statistics
total_words = len(words)
unique_words = len(word_counts)

print("\n=== Analysis Results ===")
print(f"Total words: {total_words}")
print(f"Unique words: {unique_words}")

try:
    n = input("Enter how many top words to display (default = 10): ").strip()
    n = int(n) if n else 10
    if n <= 0:
        print("Invalid number. Defaulting to 10.")
        n = 10
    if n > unique_words:
        print(f"N is larger than the number of unique words. Showing top {unique_words}.")
        n = unique_words
    except ValueError:
        print("Invalid input. Defaulting to 10.")
        n = 10

    print(f"\nTop {n} words:")
    for word, freq in get_top_n(word_counts, n):
        print(f"{word}: {freq}")

    print("\n=== End of Report ===")

if __name__ == "__main__":
    word_frequency_counter()

```

Outputs:

```

(venv) user@fedora:~/Downloads/Clash$ python wordcount.py
=== Word Frequency Counter ===
Enter the path to a text file: words.txt

=== Analysis Results ===
Total words: 6
Unique words: 6
Enter how many top words to display (default = 10): 10
N is larger than the number of unique words. Showing top 6.

```

Top 6 words:

b: 1

sdkbc: 1

cniowh: 1

ewhvwei: 1

whicweiwecn: 1

eiewopj: 1

(venv) user@fedora:~/Downloads/Clash\$ python wordcount.py

==== Word Frequency Counter ====

Enter the path to a text file: words.txt

==== Analysis Results ====

Total words: 39

Unique words: 32

Enter how many top words to display (default = 10): 10

Top 10 words:

is: 3

the: 3

my: 2

this: 2

i: 2

hi: 1

name: 1

akula: 1

charith: 1

mini: 1

==== End of Report ====