Word Counter Development

Python programming Internship Project

Submitted by: P Mukesh Varma

Date: 15/02/2025

1. Introduction

The Word Counter project is part of the Python Programming Internship.

It involves building a program that counts the number of words from user input.

This project covers input handling, string manipulation, functions, control flow,

and output display.

2. Objectives

- Learn input handling in Python
- Explore string manipulation techniques
- Create functions for modular programming
- Implement basic control flow structures
- Display word count results in a user-friendly format

3. Requirements and Features

- User Input: Supports multi-line input from the user
- Word Counting Logic: Uses a regex-based function to count words
- Output Display: Displays word count clearly on the console
- Error Handling: Catches empty input and prompts the user
- Code Comments: Provides clear explanations for each part of the code
- User-Friendly Interface: Includes prompts and outputs for ease of use

4. Python Source Code (Final and Corrected)

```
def count_words(text):
   """Counts the number of words in the given text
   using regex.""" words = re.findall(r'\b\w+\b',
   text)
   return len(words)

def main():
   print("Welcome to the Word Counter Program")
```

```
    print("Enter your text (press Enter twice to finish):")

    # Collect multi-
    line input lines
    = []
    while True:
        line =
        input()
        if line
        == "":
            break
        lines.append(l
        ine)

    text = "\n".join(lines).strip()

    # Error handling for
    empty input if not
    text:
        print("Error: No input provided.
        Please enter some text.") return

    # Count words and display
    output word_count =
    count_words(text)
    print(f"Word Count:
    {word_count}")

if __name__== "__
    main__": main()
```

5. Code Explanation

- Function Creation: `count_words()` uses regex to identify words and count them
- Multi-line Input: Allows users to input multiple lines until they press Enter twice
- Input Handling: `input()` collects user input, and empty input is managed as an error
- Output Display: Displays results clearly using `print()`
- Error Handling: Checks for empty inputs and prints error messages
- Code Comments: Provides explanations for every function and key step

6. Output Verification Examples

Example 1: Counting Words from a Single

Line Input: "Hello world from Python"

Output: "Word Count: 4"

Example 2: Counting Words from Multiple Lines

Input:

Python is fun.

It is easy to learn.

Output: "Word Count: 8"

Example 3: Handling Empty Input

Input: (no text entered)

Output: "Error: No input provided. Please enter some text."

7. Conclusion

- The Word Counter project meets all the internship requirements.
- It demonstrates key concepts such as input handling, string manipulation, and regex usage.
- The project has proper error handling and user-friendly output displays.
- It is structured using functions and comments, ensuring clarity and modularity.
- With added multi-line input and improved error management, it is practical and comprehensive