**Pseudocode:**

# Define the path to the speech file

file\_path = "state\_of\_the\_union.txt"

# Open the file and read its full text into a variable

with open(file\_path, "r") as file:

speech = file.read()

# Split the speech into sentences using punctuation (., !, ?)

sentences = split text by sentence-ending punctuation

# Extract all word tokens using regular expressions

words = find all word-like patterns in the text

# Calculate total number of words

word\_count = length of words list

# Calculate total number of characters in the original text

char\_count = length of entire speech string

# Calculate average word length

avg\_word\_len = total characters in words / word\_count

# Calculate average sentence length in words

avg\_sent\_len = word\_count / number of sentences

# Convert all words to lowercase for consistent counting

lowercase\_words = convert all words to lowercase

# Count frequency of each word

word\_freq = dictionary of word counts from lowercase\_words

# Get the 10 most frequent words

top\_words = top 10 entries from word\_freq

# Identify unique words

unique\_words = set of words

# Sort unique words by length in descending order

sorted\_by\_length = sort unique\_words by word length (descending)

# Get the top 10 longest words

longest\_words = first 10 words from sorted\_by\_length

# Display statistics in a summary table

print word count, character count, average word length, average sentence length, top 10 longest words

# Display top 10 word frequencies in another table

print top\_words and their counts

# End of program

**Source Code:**

*"""Title: State of the Union Speech Analyzer*

*Description: This Python script reads a speech from a text file and performs*

*statistical analysis including word count, character count,*

*average word and sentence lengths, word frequency distribution,*

*and the longest words used in the speech.*

*Author: Ian Patricio*

*Date: July 26, 2025*

*"""*

import re

from collections import Counter

from tabulate import tabulate

# Load the speech from a file

file\_path = "state\_of\_the\_union.txt" # Replace with your actual path if needed

with open(file\_path, "r", encoding="utf-8") as file:

speech = file.read()

# Sentence and word tokenization

sentences = re.split(r'[.!?]+', speech)

words = re.findall(r'\b\w+\b', speech)

# Basic statistics

word\_count = len(words)

char\_count = len(speech)

avg\_word\_len = sum(len(word) for word in words) / word\_count

avg\_sent\_len = word\_count / len(sentences)

# Word frequency (case insensitive)

word\_freq = Counter(map(str.lower, words))

most\_common\_words = word\_freq.most\_common(10)

# Top 10 longest unique words

unique\_words = set(words)

longest\_words = sorted(unique\_words, key=len, reverse=True)[:10]

# Summary table

summary\_table = [

["Word Count", word\_count],

["Character Count", char\_count],

["Average Word Length", f"{avg\_word\_len:.2f}"],

["Average Sentence Length (words)", f"{avg\_sent\_len:.2f}"],

["Top 10 Longest Words", ", ".join(longest\_words)],

]

# Display summary

print("\nSpeech Summary:")

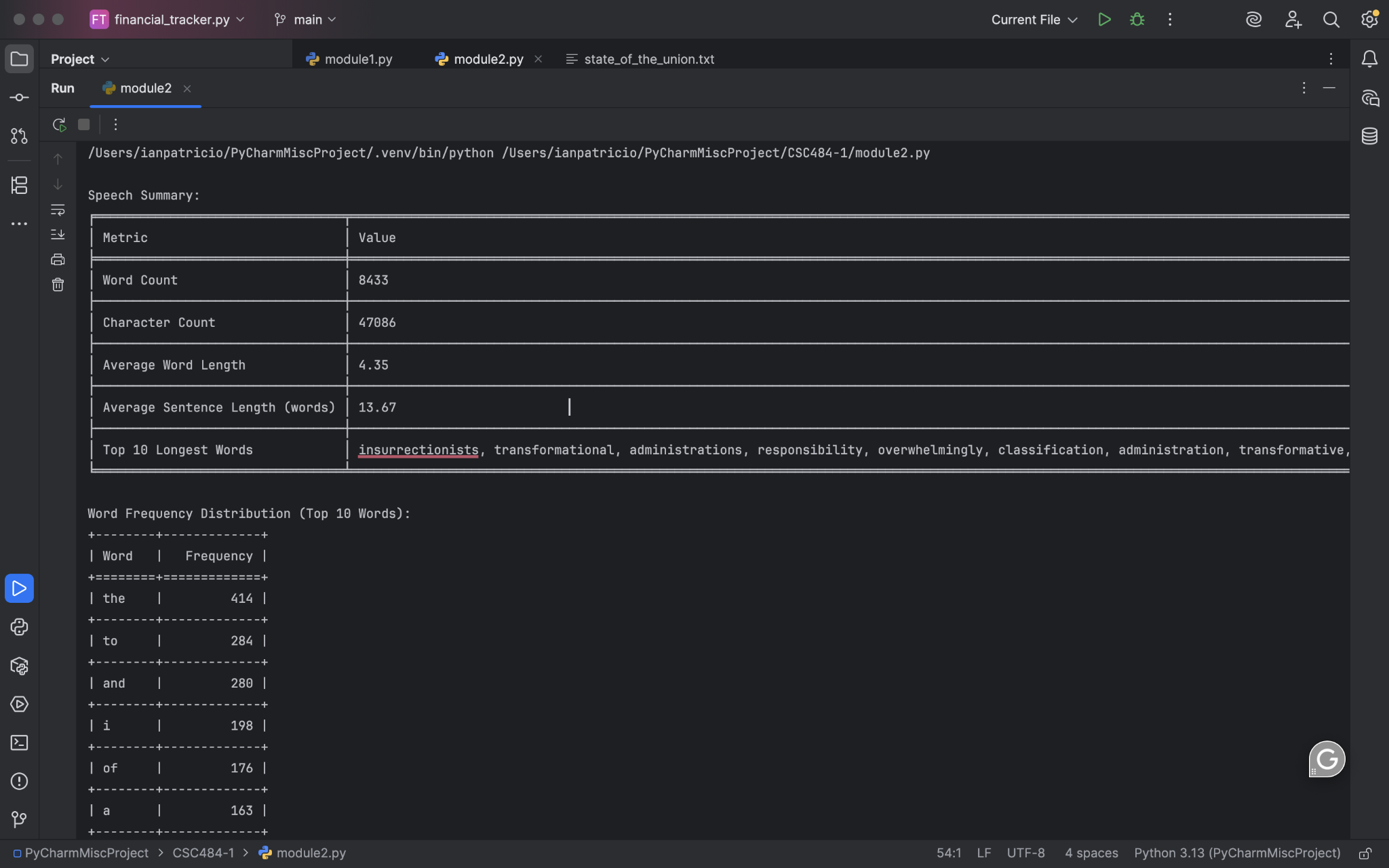
print(tabulate(summary\_table, headers=["Metric", "Value"], tablefmt="fancy\_grid"))

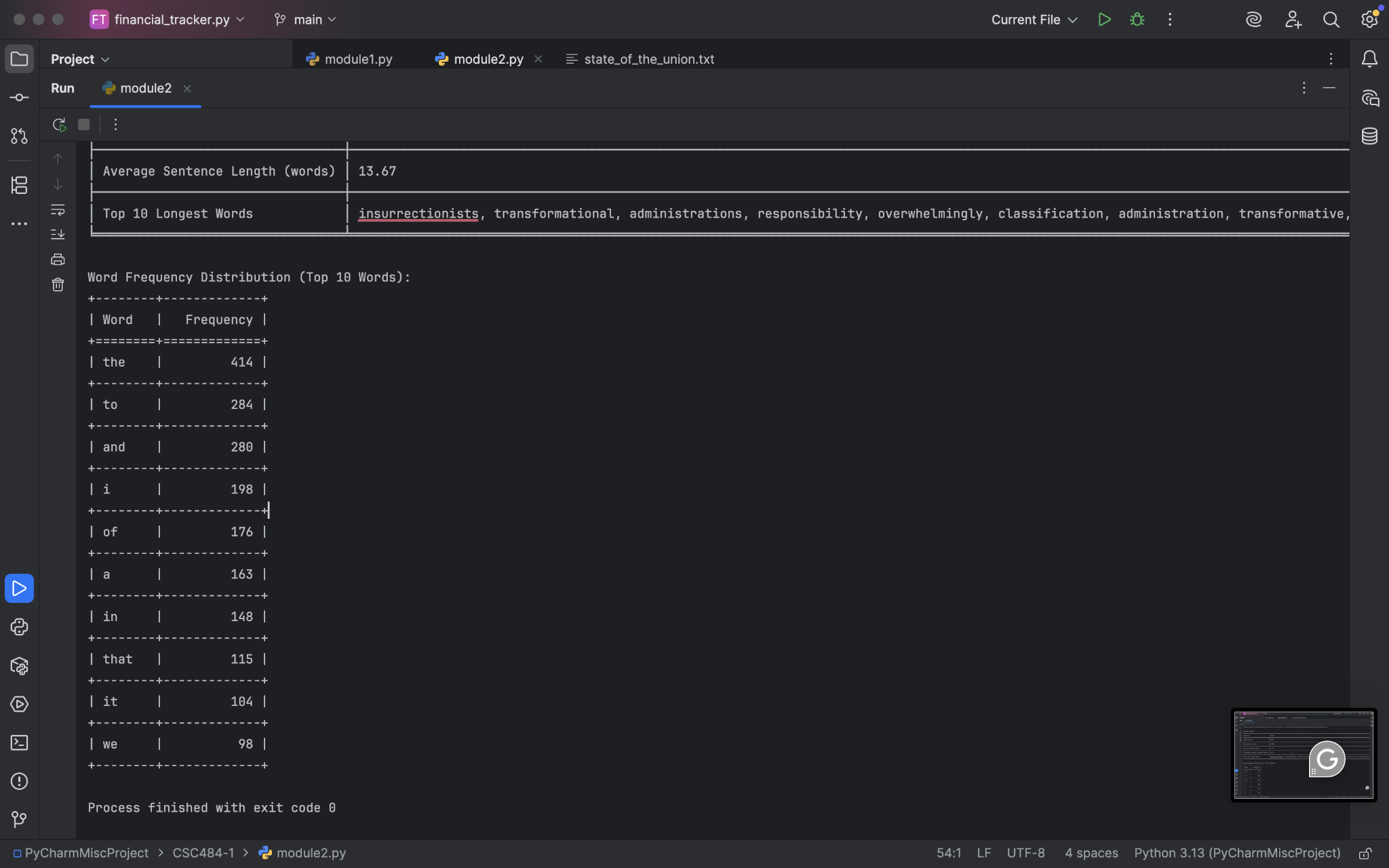
# Display word frequency

print("\nWord Frequency Distribution (Top 10 Words):")

print(tabulate(most\_common\_words, headers=["Word", "Frequency"], tablefmt="grid"))

**Screenshots:**





**Git Repository:**

https://github.com/ianpatricio-csuglobal/CSC484-1