Project Report

Spring 2025

Name: Rakshith
Project: Counting Weekends

Note: As you write each section, try to be as clear and detailed as possible. Your goal is to communicate your thought process and work clearly. Don't worry if you faced challenges or made mistakes; discussing these is a valuable part of learning and shows your problem-solving skills! Remember, there's no single 'right' way to do these tasks, so be creative and honest in your responses.

Problem Statement (2-3 Paragraphs):

The goal of this assignment was to create a Python program that counts how many weekends occur between two user entered dates. The program accepts dates in m/d/y, m/d, or y-m-d format. If only m/d is entered, the year defaults to 2026, and two digit years are treated as 2000 to 2099. The program outputs the formatted start and end dates, their day names, the number of full weekends, Saturday only weekends, Sunday only weekends, and the total weekends.

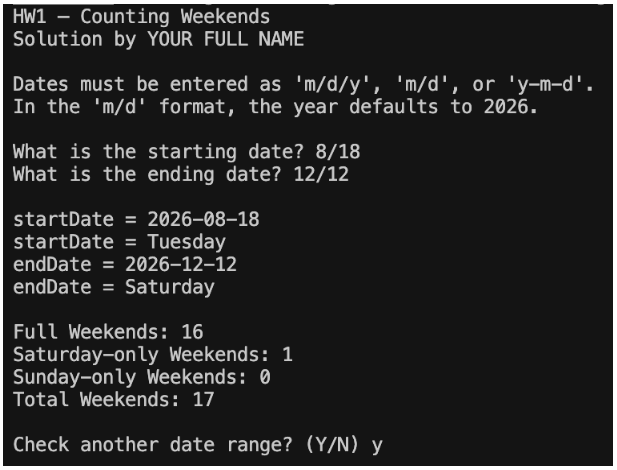
I handled errors using try-except blocks to catch invalid formats, impossible dates like 2/31/2025, and years outside 1800 to 2200. If the dates are entered in reverse order, the program swaps them automatically. The program continues running until the user chooses to stop.

Design (1-3 Paragraphs):

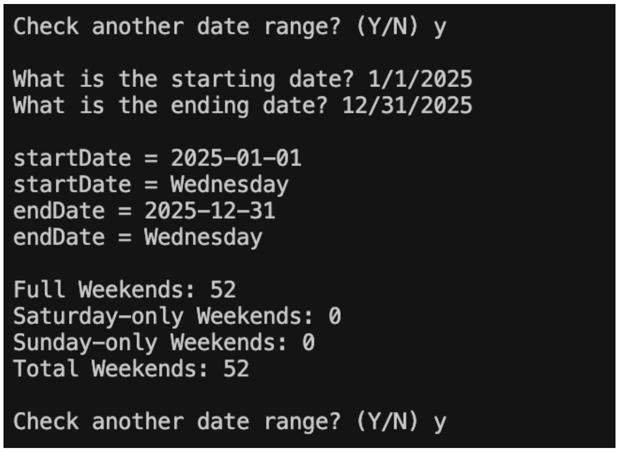
I structured the program using separate functions for parsing dates, counting weekends, and controlling the main loop. I used Python’s datetime module because it simplifies date validation and weekday detection. The counting logic moves one day at a time from the start date to the end date and checks whether each day is Saturday or Sunday. This approach is clear and reliable, and the date range limits make it efficient enough. Dividing the program into functions made the code easier to read, test, and debug.

Testing (1-2 Paragraphs + screenshots of 3 test cases):

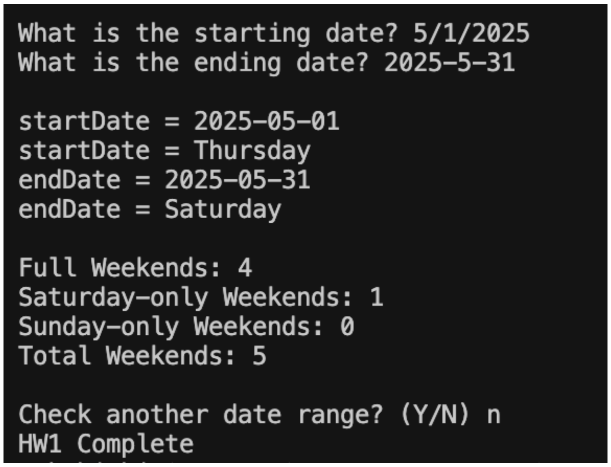
I tested normal ranges like 1/1/2025 to 12/31/2025 and shorter ranges like 5/1/2025 to 5/31/2025 to verify correct weekend counts. I also tested reversed dates, single day ranges, and invalid inputs such as 2/31/2025 to confirm that error handling worked correctly. The program produced accurate results and handled invalid entries without crashing.



```
HW1 – Counting Weekends
Solution by YOUR FULL NAME

Dates must be entered as 'm/d/y', 'm/d', or 'y-m-d'.
In the 'm/d' format, the year defaults to 2026.

What is the starting date? 8/18
What is the ending date? 12/12

startDate = 2026-08-18
startDate = Tuesday
endDate = 2026-12-12
endDate = Saturday

Full Weekends: 16
Saturday-only Weekends: 1
Sunday-only Weekends: 0
Total Weekends: 17

Check another date range? (Y/N) y
```



```
Check another date range? (Y/N) y

What is the starting date? 1/1/2025
What is the ending date? 12/31/2025

startDate = 2025-01-01
startDate = Wednesday
endDate = 2025-12-31
endDate = Wednesday

Full Weekends: 52
Saturday-only Weekends: 0
Sunday-only Weekends: 0
Total Weekends: 52

Check another date range? (Y/N) y
```



```
What is the starting date? 5/1/2025
What is the ending date? 2025-5-31

startDate = 2025-05-01
startDate = Thursday
endDate = 2025-05-31
endDate = Saturday

Full Weekends: 4
Saturday-only Weekends: 1
Sunday-only Weekends: 0
Total Weekends: 5

Check another date range? (Y/N) n
HW1 Complete
```

Conclusion (1 paragraph)

This project improved my understanding of input validation and working with dates in Python. Organizing the program into functions helped keep the logic clear and manageable. The program met all requirements and performed correctly across different test cases, and I would continue using this structured approach in future assignments.