**Task 03**

**Password Complexity Checker**

**Build a tool that assesses the strength of a password based on criteria such as length, presence of uppercase and lowercase letters, numbers, and special characters. Provide feedback to users on the password's strength.**

## ****📘 Introduction****

In today’s digital age, **password security** is one of the most crucial aspects of protecting user data. Weak passwords are often the first vulnerability exploited by attackers. This project presents a **Password Strength Checker Tool**, built using Python, which evaluates the strength of a given password based on standard security criteria.

This tool provides immediate feedback to users to help them create **stronger and more secure passwords**. It is ideal for educational purposes, login systems, or security-awareness applications.

## ⚙️ Working of the Tool

The tool works by:

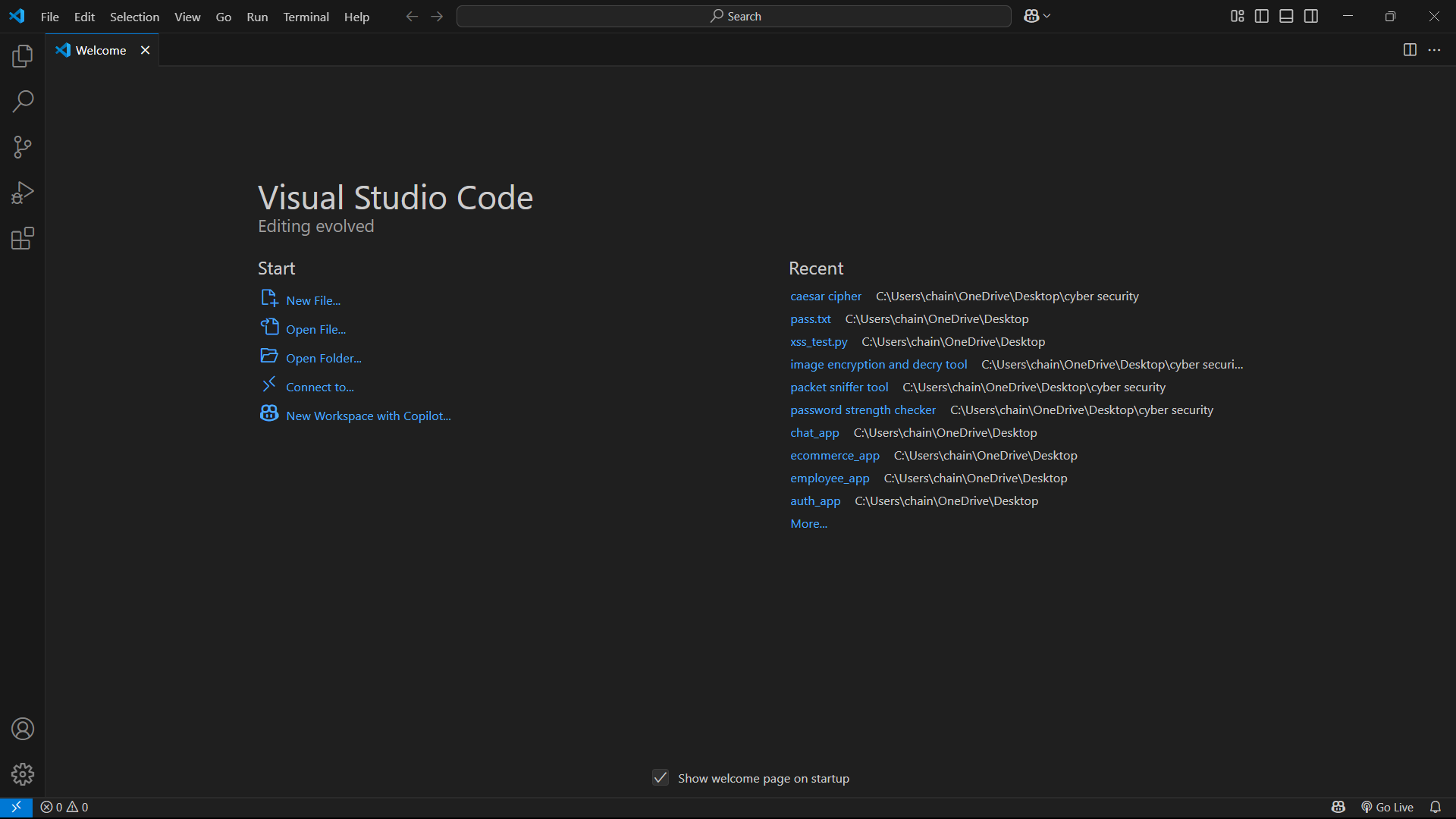
1. Prompting the user to input a password.
2. Checking the password against the following criteria:
   * Length (at least 8 characters)
   * Presence of lowercase letters
   * Presence of uppercase letters
   * Presence of digits
   * Presence of special characters (e.g., !@#$%^&\*())
3. Scoring the password based on how many criteria it satisfies.
4. Displaying a strength rating (Very Weak, Weak, Moderate, Strong, Very Strong).
5. Providing suggestions for improvement if the password is not strong enough.

**Implementation of code in Python**

**Steps: -**

1. Choose a code editor or IDE as per your choice.
2. Install and set up **VS code.**

Download: Visit <https://code.visualstudio.com/> and install VS Code.



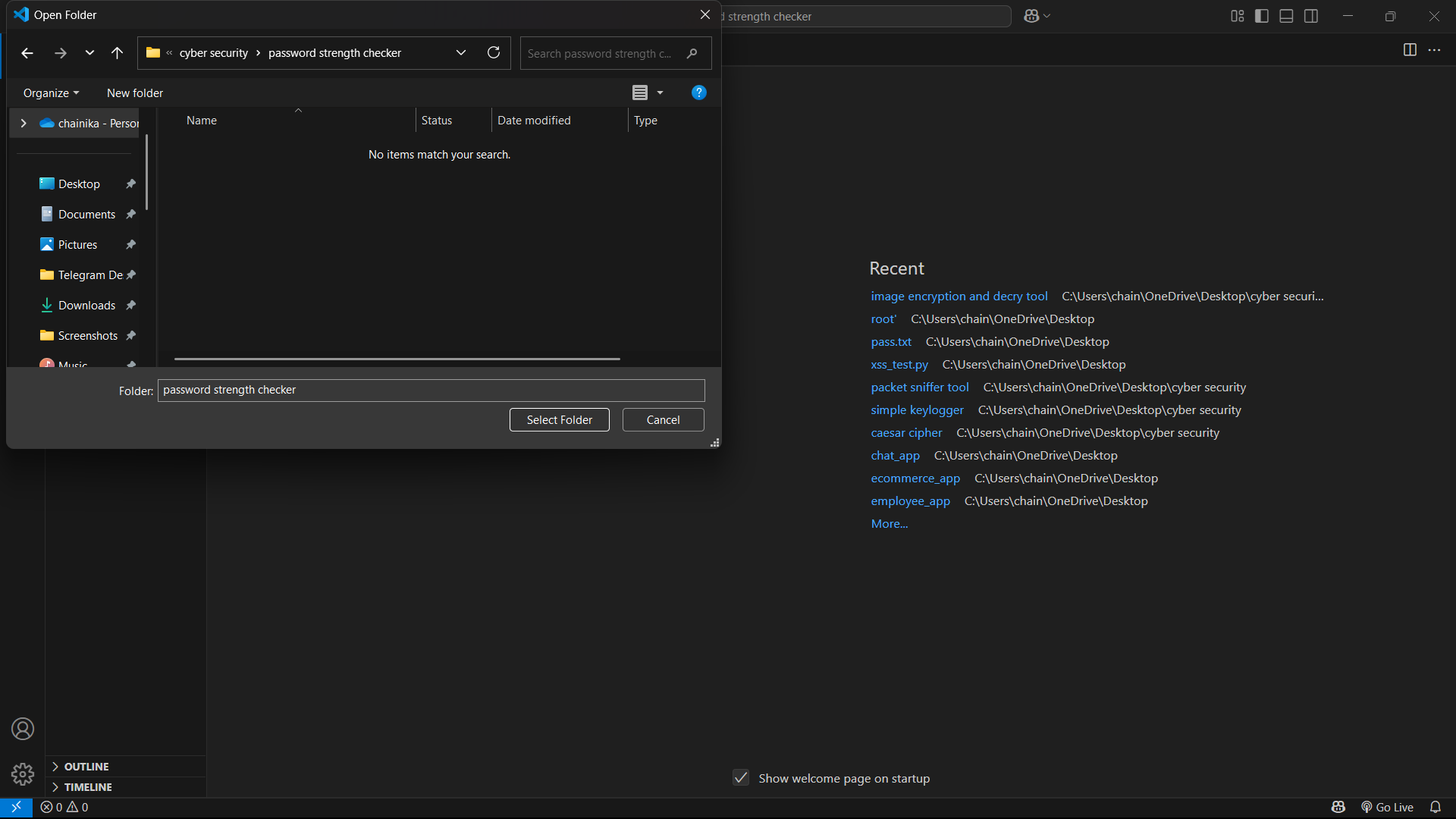
Go to extension tab (ctrl+ shift+ x) and search for Python. Click install on the official Microsoft Python extension.

Choose python interpreter (I have selected Python 3.12.7)

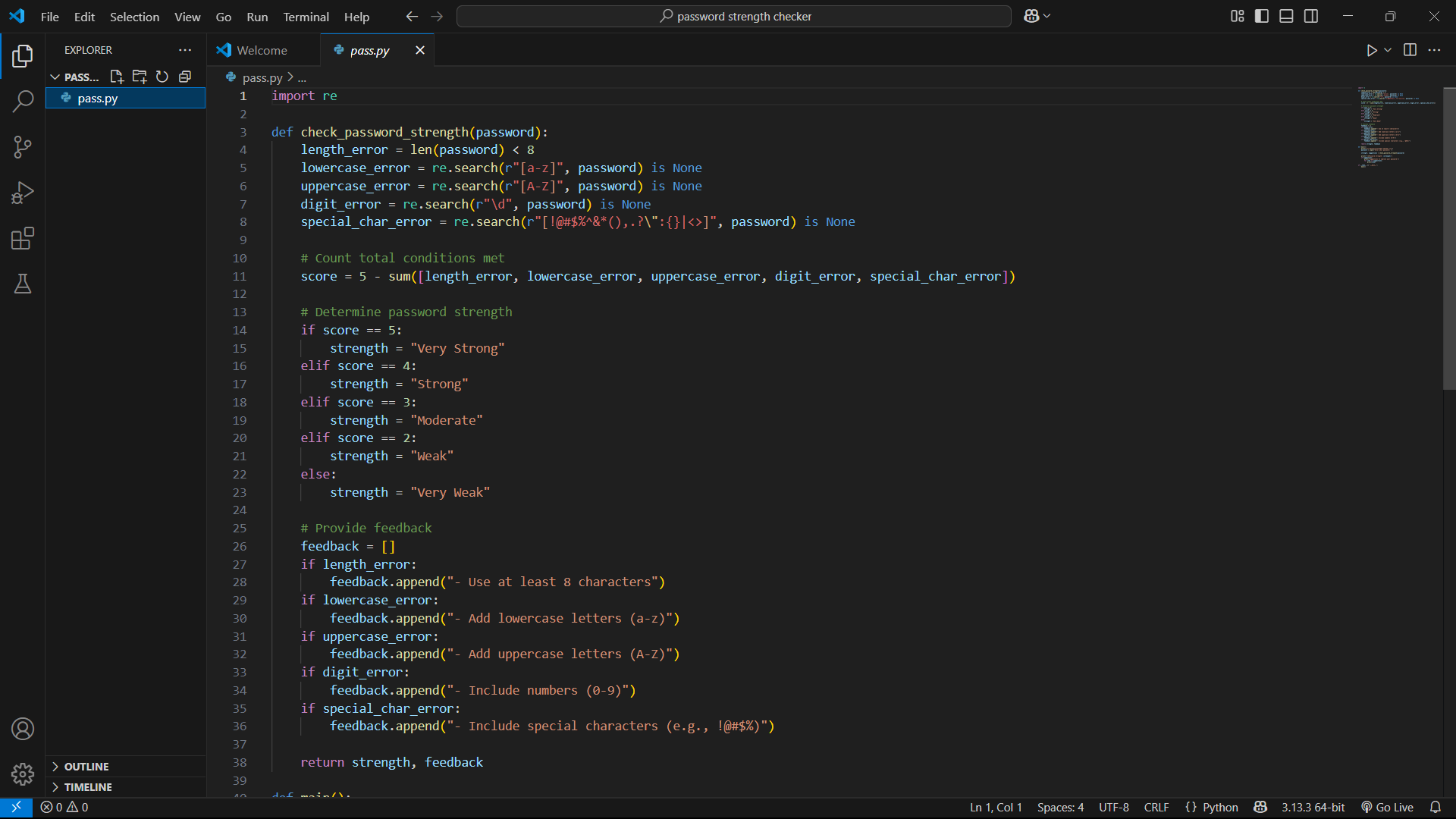
1. Create and save a python file (file having extension .py)

In VS Code: File : New File.

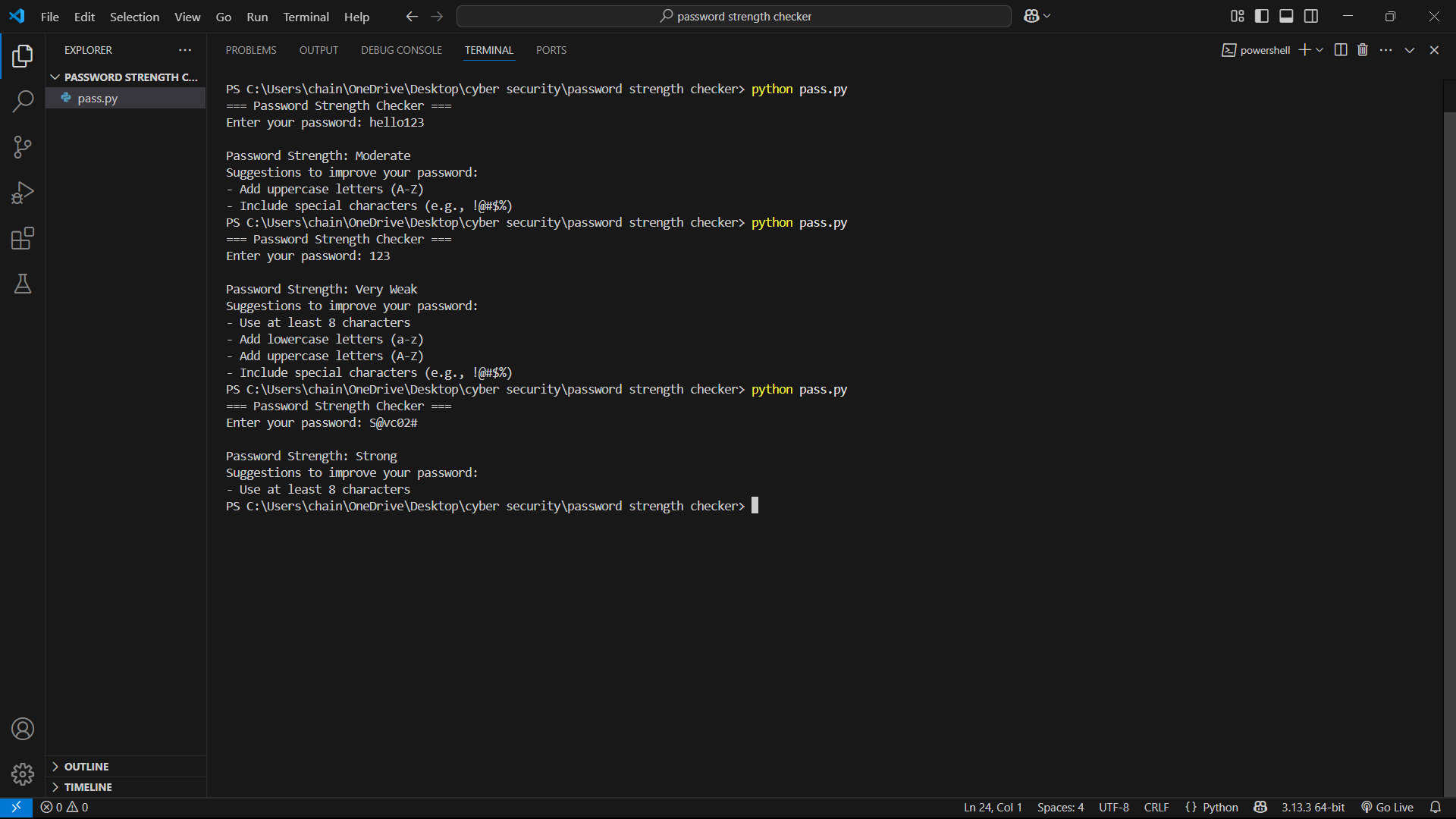
And, Save the file extension .py



1. Initiate with writing your program:



1. Then Run the program in the terminal:



Successful running of program with the result shows the correct program.

## ****🎯 Significance of the Tool****

* **User Awareness**: Helps users understand how to create more secure passwords.
* **Security Best Practices**: Promotes compliance with secure password policies.
* **Real-world Application**: Can be integrated into login or registration systems.
* **Educational Value**: A great beginner-level project to understand regex, input handling, and logic building in Python.

## ✅ Conclusion

This tool is a simple yet powerful example of how basic programming can help address real-world security concerns. With a few enhancements like GUI support or web integration, it can evolve into a fully usable module in any software application.

**Presented by: - Chainika Dongre Date: - 18 May 2025**