Problem definition and analysis:

This project is a simple login and weather-based clothing suggestion system. The problem I wanted to solve was how to create a program that allows a user to log in securely and then gives clothing advice based on the weather. I chose this idea because it uses basic Python skills I've learned, such as if statements, loops, functions, and user input. I also chose Python because I've been using it in my course and feel comfortable with it.

The goal is to let a user log in using a username and password (with up to three tries). If the login is successful, they can enter the weather (temperature, rain, and wind), and the program will suggest what clothes to wear. The program is made for any user who wants a simple way to get clothing advice based on the weather.

To plan the project, I thought about the main steps: login, getting the weather, and giving suggestions. Each step depends on the one before it. I researched by looking at class notes and small examples online to help me understand how to check passwords and give advice based on different conditions.

Documented design

The program has two major parts: the login system and the clothing recommender system. It uses straightforward Python features such as input statements, if-else statements, while loops, and functions. The design stresses simplicity, readability, and ease of use.

System Overview

The program starts by asking the user to login with a username and password. If they enter wrong information, they have three chances. Once they have logged in successfully, a menu appears with three options:

1. Input today's weather
2. Get clothing suggestion
3. Exit program

If the user chooses to enter the weather, the user is asked to enter the temperature, whether it is raining, and whether it is windy. If they choose to get a clothing tip, the program gives a tip based on the weather the user enters.

Main Modules

- The login function uses a loop to make sure the username and password are correctly entered. It tracks how many tries the user has attempted logging in.
- The main menu is shown after a successful login. The application is waiting for the user to make a choice.

- The weather input section asks the user to enter three things: temperature in Celsius, raining (yes or no), and windy (yes or no). It checks that the user enters correct responses.
- The dress suggestion part verifies the weather input and suggests tips. If cold, it suggests a thick coat. If warm, it suggests a t-shirt. If raining, it suggests an umbrella. If windy, it suggests windproof clothing.

Data and Variables

The program stores the correct username and password as constants. It monitors the attempts made in logging in. It stores the temperature, rain status, and wind status after the user has input them. It uses a flag to check whether weather details have been input before previously allowing clothing suggestions.

Algorithms

For logging in, the app uses a while loop with a limit of three attempts. If the user enters the correct username and password, it logs him in and shows him a success message. If they enter it incorrectly three times, it shows an error and stops.

The menu is shown with a loop that runs indefinitely until the user wants to exit. When the user chooses option 1, he goes ahead and inputs the weather. If he chooses option 2, the program only makes a suggestion if weather is input first. Option 3 closes the program.

Interface Design

All that is done with the program is text input and output to the terminal. The user types in their responses, and the program responds with messages and suggestions. It gives clear instructions for each step. For example, it asks for "yes" or "no" responses and double-checks the input.

Examples of suggestions are:

- "Bring a heavy coat" for cold weather.
- "Bring light layers" for mild weather.
- "Wear a t-shirt and shorts" when it is warm.
- "Bring an umbrella" when it rains.
- "Bring wind-resistant clothes" when it is windy.

Libraries Used

No libraries are used in this project. Everything is done using standard Python features like input, print, and standard control structures.

Error Handling

If the user has entered a too high or low temperature to make sense, the program asks the user to enter it again. The same applies to rain and wind — the program only lets the user input "yes" or "no". If the user tries to get a suggestion before entering the weather, the program tells him/her to enter the weather first.

Testing and evaluation:

To make sure the program works correctly, I tested each part of it separately. I first tested the login system by entering the correct and incorrect usernames and passwords. It allowed access only with the right details and blocked access after three wrong tries, just as planned.

Next, I tested the main menu. I made sure that each option works: entering the weather, getting suggestions, and exiting the program. I also tested what happens if the user chooses an invalid option, and the program showed a warning as expected.

For the weather input, I tested different temperature values. If I entered a number too high (like 70°C) or too low (like -50°C), the program asked me to try again. I also tested typing other words instead of “yes” or “no” for rain and wind, and the program handled it correctly.

When I tested the clothing suggestion option, the program gave advice based on the weather data I entered. If I didn’t enter weather info first, it told me to go back and do that.

Overall, the program met all my original goals: working login, weather input, and clothing advice. If I improve it in the future, I would add more detailed advice or a graphical interface.