Lab Printing System

You are asked to write a program that simulates a Lab Printing System. As a project, it serves two purposes: The program is designed to give you some practice writing programs that manipulate strings, data structures and files, as well as apply sorting and searching algorithms.

The computer first displays a login form. There are two user types in this system:

- **Admin**
- **Users**

If the user enters "**admin**" as username and "**admin123123**" as password, the system will show the admin menu. Else, if a normal user logins without password (empty password), a different menu will be shown.

If the admin inputs an incorrect combination, the system should display "Incorrect Username and/or Password". The admin is not allowed to enter more than **5 wrong combinations**. Otherwise, the menu of the system will be displayed.

The System

Your job is to write a script that handles the user interaction component of the system. To solve the problem, your program must be able to:

- Read one text file and upload all the jobs into the special queue of the system
- Implement the basic control structure and manage the details

The **text file** includes several printing jobs in the following format:

job101, charbel, 20220802, 0
job102, moe, 20220803, 2
….
This file represents the list of printing jobs which were issued. Each line contains 4 words: job id, username, timestamp (YYYYMMDD), and the last word represents the priority (integer). You are asked to import that file when the program starts. You have to remove duplicates from the file (same usernames are not allowed).

The Flow of The System:
A. The jobs written inside the text file are imported (loaded) into the Special Queue of the program with no intervention and without letting the user know anything.
B. The program starts by greeting the user and asking for their username and password (if the password was left empty, then it's a normal user)

If the user logged in is an admin, the following menu will be displayed:

1. **Display Statistics**
2. **Add a Job**
3. **Display all Jobs**
4. **Change Job's Priority**
5. **Remove Job**
6. **Run Printer**
7. **Exit**

- If the admin chooses (1), the system should display **how many jobs** are to be printed **today**.
- If the admin chooses (2), the system should allow the admin to **add a new job** to the printer by specifying the **username, and the priority only** (the job id is **auto-incremented** and **the date is automatically taken** from the computer)
- If the admin chooses (3), the system should display **all the jobs registered in the system ordered by date (Today, Tomorrow, etc). Don't show old jobs.**
- If the admin chooses (4), the system should allow the admin to **change the priority** of a job by specifying the Job Id. The system should prompt the admin for the Job ID and for the priority. **If the job is found**, then the priority is changed.
- If the admin chooses (5), the system should allow the admin to **remove a job** from the system by **asking for the ID**
- If the admin chooses (6), the system should run **today's jobs found inside the queue**. The system should display the jobs on the screen (**sorted by their priority**) and remove them from the queue.
- If the user chooses (8), **the program should exit without saving.**

If the user logged in is a normal user, the following menu will be displayed:

1. **Add a new Job**
2. **Exit**

- If the user chooses (1), the system should **add a new job to the system** by taking th**e username from the login** system, the current date of the job, and the **priority by default is 0**. (The job id is automatically incremented)
- If the admin chooses (2), the program should **terminate by saving the new added job in the text file**.

After each option, the menu is displayed again. The user can use the system multiple times.