Project Title: Password Manager and Generator

Project Description:

The **Password Manager and Generator** is a security tool designed to generate, validate, and manage strong passwords for users. It provides an automatic way to create secure passwords while ensuring that they meet various strength criteria. The application also allows users to check whether a password is commonly used or potentially vulnerable by comparing it to a list of common passwords.

Key Features:

1. **Password Generation:**
    - The tool generates a random password based on user-defined criteria such as length, inclusion of uppercase letters, digits, and special characters.
    - The generated password includes lowercase letters by default, and users can optionally include uppercase letters, digits, and special characters.
    - The password is built using a secure random selection from a predefined set of characters, ensuring that the passwords are complex and difficult to guess.
2. **Password Validation:**
    - The tool validates the generated password by checking if it meets strength requirements, which include:
        - A minimum length of 8 characters
        - At least one uppercase letter
        - At least one lowercase letter
        - At least one digit
        - At least one special character (e.g., !@#$%^&*()-_=+<>?)
    - The password is only considered strong if it meets all these conditions.
3. **Common Password Check:**
    - To enhance security, the tool checks whether the password is commonly used by referencing a list of common passwords (common_pwds.txt).
    - The list of common passwords can be easily updated or expanded to include newly discovered commonly used passwords.
    - This check helps users avoid using passwords that are easy to guess and commonly targeted by attackers.
4. **User Interaction:**
    - The user can run the tool to generate a new password and immediately check its strength and whether it’s common or not.
    - The tool outputs feedback on the strength of the password and provides guidance:

        - If the password is strong and unique, it confirms that the password is secure.
        - If the password is common, it warns the user against using it.
        - If the password fails to meet strength requirements, it provides feedback on which criteria are missing.

5. **File Handling:**
    - The tool reads the common_pwds.txt file, which contains a list of common passwords, to check against the user's generated password.
    - The application handles potential I/O exceptions when reading the file, ensuring robustness in case the file is missing or inaccessible.

Technologies Used:

- **Java Programming Language:** The application is built in Java, utilizing its built-in libraries such as java.util.Random for randomization and java.nio.file.Files for reading files.
- **Regular Expressions (Regex):** Regular expressions are used to define and check password strength criteria, providing a flexible and efficient way to validate passwords.
- **File I/O:** The tool interacts with external files for reading common passwords, allowing easy updates and maintenance.