HOME OTHERS

# SECUREPASS:PASSWORD MANAGER & GENRATOR

→

# LIST OF CONTENTS

# INTRODUCTION

This project is a password manager and generator built with Python using Tkinter. It allows users to generate, hash, and verify the strength of passwords, as well as generate QR codes for secure storage.

Key Features:
* Generate passwords
* Verify password strength
* Hash passwords using SHA256
* Generate QR codes for passwords
* Multilingual suppor

# CREATING THE INTERFACE WITH TKINTER

Details:
* Used Tkinter to create the window with a size of 1000x850 px and a custom background color.
* Implemented a clean design with buttons, labels, and input fields to allow for user interaction.
* Integrated a language selector for English and French.

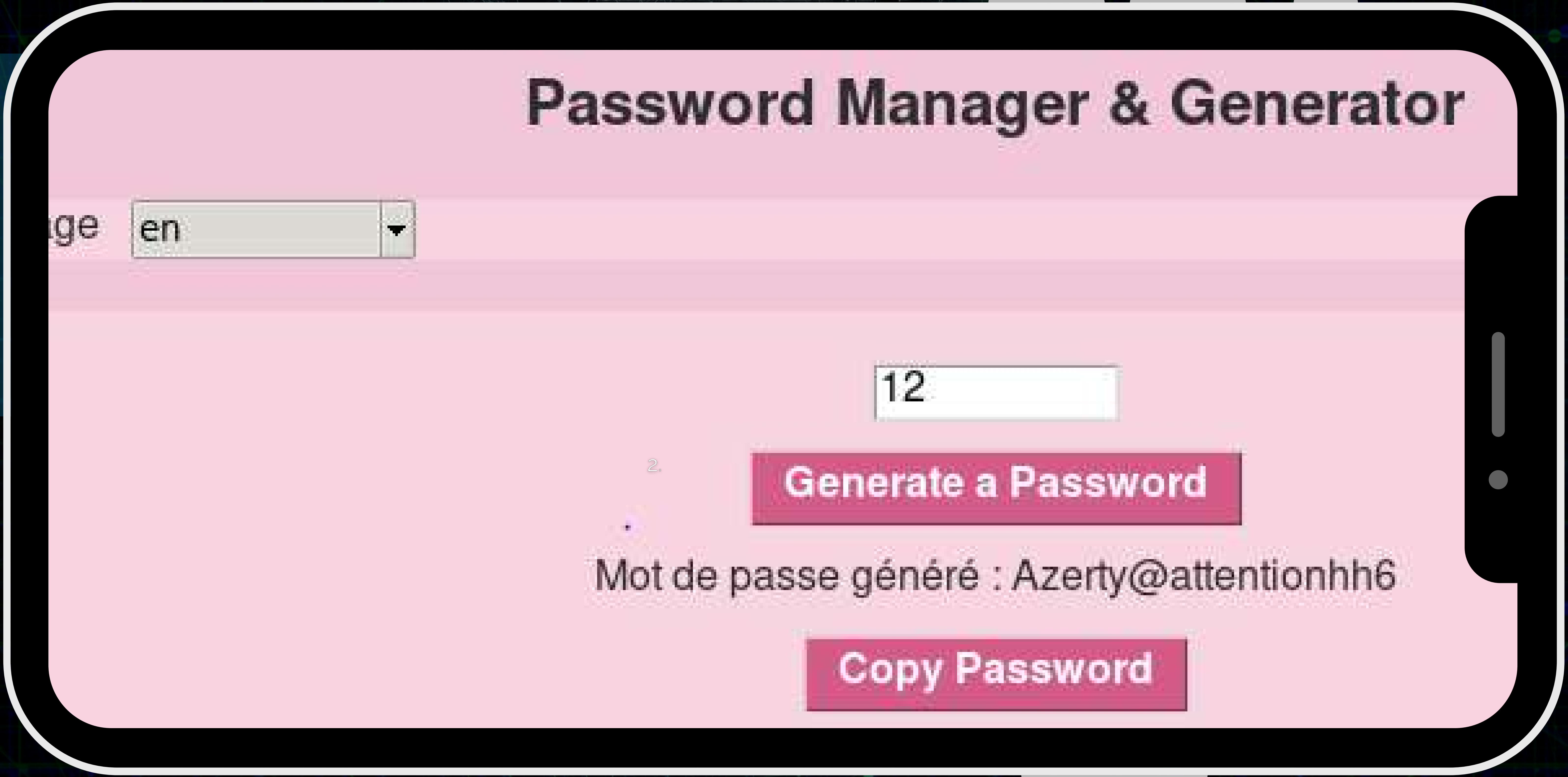
Challenges:

- One difficulty was ensuring that the interface was both functional and aesthetically pleasing. I had to carefully manage the layout and style properties to make it intuitive for users.

# GENERATING SECURE PASSWORDS

* Details:
* The password generator uses a list of common words (e.g., tree, moon) and randomly combines them with special characters and numbers. * The result is a secure, memorable password. * Users can specify the length of the generated password.

* Challenges: Initially, I wanted to give a random hash for each password, but that led to problems. The generated passwords were not usable because they were too complex. I had to focus on creating a strong, readable password format instead.



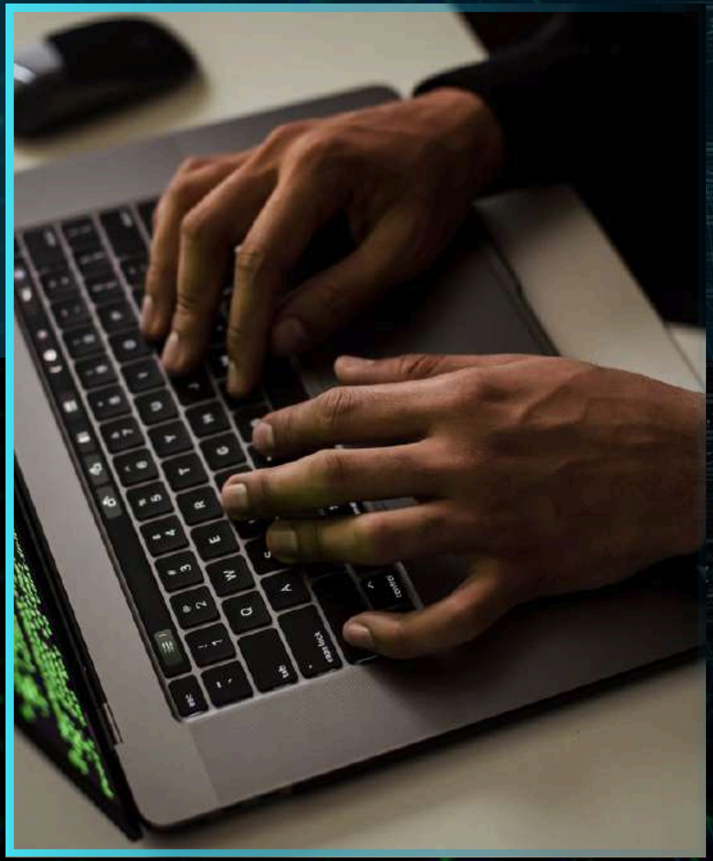
Password Manager & Generator

ge en

12

Mot de passe généré : Azerty@attentionhh6

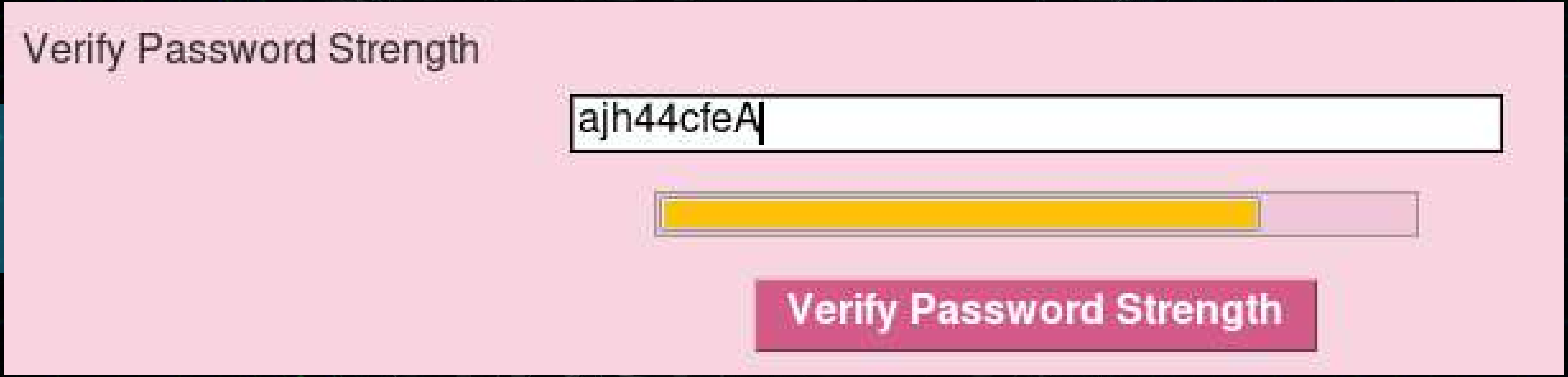
# VERIFYING PASSWORD STRENGTH



* The app checks the password strength based on length, use of uppercase and lowercase letters, digits, and special characters.
* A progress bar indicates the password's strength with different colors for weak (red), medium (yellow), and strong (green) passwords.

Challenges:

- One issue was updating the progress bar dynamically while also ensuring that the password strength evaluation was accurate. This required carefully setting up event triggers and updating the UI in real time.

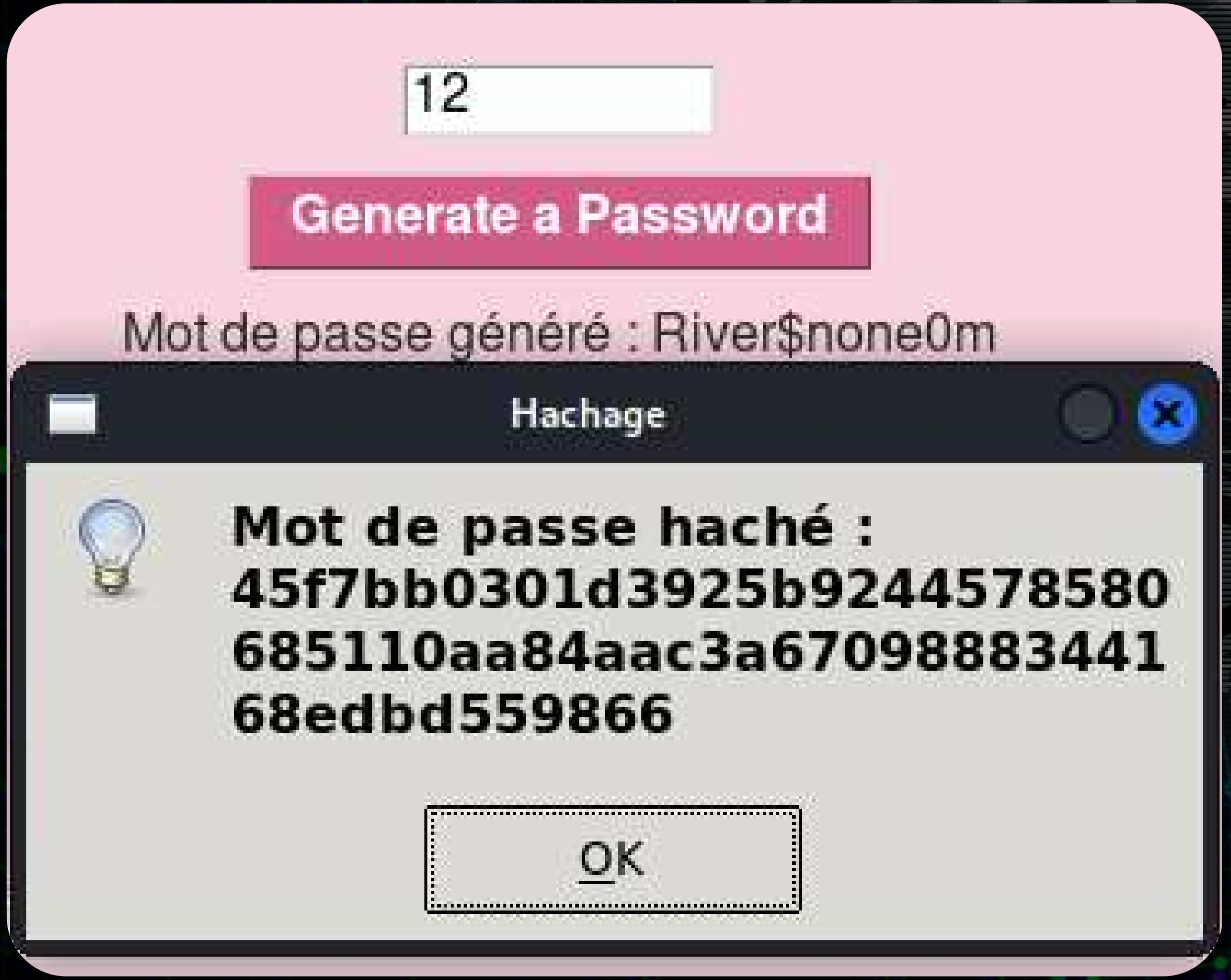


Verify Password Strength

ajh44cfeA

# HASHING THE PASSWORD FOR SECURITY

* Added the option to hash the generated password using SHA256 for secure storage.
* This converts the password into a fixed-length string that is not reversible, enhancing security.



12

```
Mot de passe généré : River$none0m
```

```
Mot de passe haché :
45f7bb0301d3925b9244578580
685110aa84aac3a67098883441
68edbd559866
```

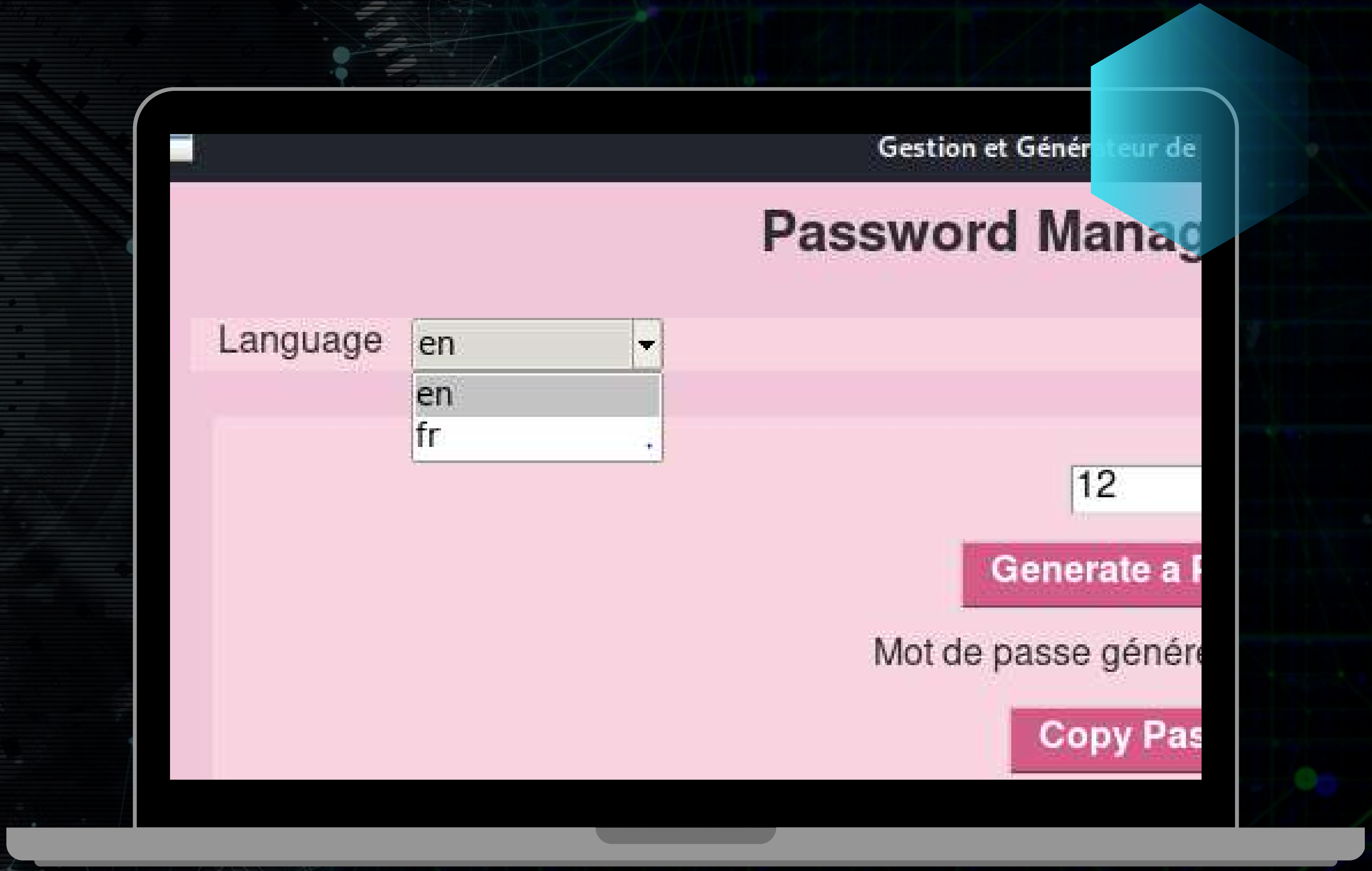
* Challenges:

- Initially, I faced issues when trying to hash any password without considering that certain passwords might be too simple. I later realized that I needed to ensure the passwords were strong enough before hashing them.

# SUPPORTING MULTIPLE LANGUAGES

* Integrated an English/French language toggle to support a wider user base.

* The interface dynamically updates based on the selected language.



Gestion et Génér teur de

Password Manag

Language en

# CONCLUSION

Building this project improved my understanding of working with Tkinter and handling user inputs. It also deepened my knowledge of password security practices

LICERIA & CO.

HOME ABOUT CONTENT OTHERS

# THANK YOU

www.reallygreatsite.com