Solution authentification forte :

Obligation d'avoir un mot de passe robuste (12 caractères minimum, avoir au moins une majuscule, minuscule, chiffre et caractère spécial.) :

Pour cela, lors de la création du mot de passe il faut :

- Pour les 12 caractères : vérifier que la longueur du mot de passe choisi soit supérieure ou égale à 12.
- Importer le module « re » qui va permettre de chercher des expressions dans une chaîne. On va utiliser grâce à ce module « re.search(pattern, string) » où pattern est l'expression (par exemple ''[A-Z]'' pour cherche une majuscule) et string est la chaîne dans laquelle on cherche l'expression.

Système d'authentification à double facteurs :

Après avoir entré le bon mot de passe et bon identifiant, demander d'entrer un code généré sur l'application TOTP Google Autentificator.

Pour cela, il faut :

- Importer le module « pyotp »

- Générer une clé secrète avec pyotp.random_base32()

- Créer l'URI de l'utilisateur avec la méthode « .provisioning_uri() »

- Vérifier que l'utilisateur a saisi le bon code

URI est une chaîne de caractères standardisée qui contient toutes les informations nécessaires pour que Google Authenticator ou une autre application TOTP sache : quelle application l'utilisateur utilise, son nom d'utilisateur, sa clé secrète,… Elle a transmettre toutes les informations nécessaires à l'application d'authentification pour qu'elle puisse générer les codes.

Petit schéma explicatif : Mot de passe -> URI -> Google Authentificator -> code secret -> vérification Python

```python
import pyotp
import qrcode

# --- 1. Générer une clé secrète unique par utilisateur (stockée en base, jamais changée) ---
secret = pyotp.random_base32()
print("Clé secrète de l'utilisateur :", secret)

# --- 2. Générer une URL au format compatible Google Authenticator ---
uri = pyotp.totp.TOTP(secret).provisioning_uri(
    name="user@example.com",   # identifiant de l'utilisateur
    issuer_name="MonAppSecure" # nom de ton service
)

print("URI provisioning:", uri)

# --- 3. Générer le QR code à scanner avec l'appli Google Authenticator ---
qrcode.make(uri).show()

# --- 4. Vérification (lors de la connexion de l'utilisateur) ---
totp = pyotp.TOTP(secret)
user_code = input("Entrez le code de votre app Authenticator: ")

if totp.verify(user_code):
    print("✅ 2FA validé, accès autorisé")
else:
    print("❌ Code invalide, accès refusé")
```