

Documentație Aplicație

Creare fișier ZIP

Introducere

Aplicația 'ZipXpert' este un instrument ușor de utilizat pentru a crea arhive ZIP din fișiere și foldere selectate. Aceasta include opțiunea de a adăuga parolă pentru arhivele ZIP și de a ajusta nivelul de compresie. Aplicația este construită folosind Python cu biblioteci precum 'tkinter' pentru interfața grafică și 'pyminizip' pentru criptarea arhivelor.

Tehnologii utilizate

Tehnologiile utilizate în dezvoltarea acestei aplicații sunt:

- Python: Limbajul de programare principal.
- Tkinter: Bibliotecă pentru crearea interfeței grafice.
- pyminizip: Bibliotecă pentru comprimarea fișierelor și adăugarea de parole.
- customtkinter: Extensie pentru tkinter care permite personalizarea interfeței grafice.
- zipfile: Biblioteca standard Python pentru manipularea fișierelor ZIP.

Funcționalități

Aplicația dispune de următoarele funcționalități:

1. Selectare fișiere: Permite utilizatorului să selecteze fișiere individuale pentru a fi incluse în arhivă.
2. Selectare folder: Permite utilizatorului să selecteze un folder întreg pentru arhivare.
3. Creare arhivă ZIP: Utilizatorul poate crea un fișier ZIP cu opțiuni de compresie și parolă.
4. Ștergere fișiere selectate: Permite eliminarea fișierelor selectate din lista de arhivare.
5. Niveluri de compresie: Aplicația suportă două niveluri de compresie: 'Fără compresie (rapid)' și 'Compresie maximă'.
6. Parolă pentru arhivă ZIP: Utilizatorul poate seta o parolă pentru arhiva ZIP, oferind un nivel suplimentar de securitate.

Detalii ale implementării

Setări inițiale și interfață grafică

Aplicația folosește customtkinter pentru a crea o interfață grafică modernă, cu o temă întunecată (dark mode) și o temă de culoare albastră.

Cod:

```
ctk.set_appearance_mode('dark')
ctk.set_default_color_theme('blue')
```

Selectarea fișierelor și folderelor

Aplicația permite utilizatorilor să selecteze fișiere individuale sau un întreg folder. Fișierele din folder sunt adăugate automat la lista de fișiere pentru arhivare.

Selectarea fișierelor:

Aplicația permite utilizatorilor să aleagă mai multe fișiere pentru arhivare.

Cod:

```
def select_files():

    files = tkinter.filedialog.askopenfilenames(title="Selectează fișierele pentru arhivă")

    if files:

        add_files_to_list(files)

    else:

        tkinter.messagebox.showwarning("Selectie fișiere", "Nu ai selectat niciun fișier!")
```

Selectarea folderelor:

Permite utilizatorilor să selecteze un folder întreg, iar aplicația adaugă toate fișierele din acel folder în lista de fișiere pentru arhivare.

Cod:

```
def select_folder():

    folder = tkinter.filedialog.askdirectory(title="Selectează un folder pentru arhivă")

    if folder:

        files = []

        for root_dir, _, filenames in os.walk(folder):

            for filename in filenames:
```

```
    files.append(os.path.join(root_dir, filename))

add_files_to_list(files)

else:

    tkinter.messagebox.showwarning("Selectie folder", "Nu ai selectat niciun folder!")
```

Adăugarea fișierelor în listă

Fișierele selectate sunt adăugate în lista de fișiere care va fi afișată în aplicație. Dacă fișierul este deja în listă, nu va fi adăugat din nou.

Cod:

```
def add_files_to_list(files):

    for file in files:

        if file not in file_list: # Evităm adăugarea dupliilor

            file_list.append(file)

            file_label = ctk.CTkLabel(file_frame, text=file, anchor="w")

            file_label.pack(fill="x", padx=5, pady=2)
```

Ștergerea fișierelor din listă

Aplicația oferă utilizatorilor posibilitatea de a șterge fișierele selectate din lista de fișiere adăugate, resetând lista de fișiere.

Cod:

```
def clear_file_list():

    global file_list

    file_list = []

    for widget in file_frame.winfo_children():

        widget.destroy()
```

Crearea arhivei ZIP

Aplicația permite utilizatorilor să creeze arhive ZIP din fișierele selectate. Utilizatorul poate alege și nivelul de compresie dorit, precum și dacă dorește să adauge o parolă.

Crearea arhivei ZIP fără parolă:

Fișierele selectate sunt adăugate într-o arhivă ZIP utilizând biblioteca zipfile.

Cod:

```
def create_zip():

    if not file_list:

        tkinter.messagebox.showwarning("Selecție fișiere", "Nu ai selectat fișiere pentru arhivă!")

        return

    zip_path = tkinter.filedialog.asksaveasfilename(defaultextension=".zip", filetypes=[("ZIP files",
    "*.zip")], title="Salvează arhiva ZIP")

    if not zip_path:

        tkinter.messagebox.showwarning("Salvare fișier", "Nu ai specificat un nume pentru fișierul
ZIP!")

        return

    compression = compression_level.get()

    if compression == "Fără compresie (rapid)":

        compression_method = ZIP_STORED

    else:

        compression_method = ZIP_DEFLATED

    try:

        with ZipFile(zip_path, 'w', compression=compression_method) as zip_file:

            for file in file_list:

                zip_file.write(file, arcname=os.path.basename(file))

        tkinter.messagebox.showinfo("Succes", f"Arhivarea s-a efectuat cu succes la {zip_path}")

    except Exception as e:

        tkinter.messagebox.showerror("Eroare", f"A apărut o eroare: {e}")


```

Crearea arhivei ZIP cu parolă:

Dacă utilizatorul a ales să adauge o parolă, aplicația folosește pyminizip pentru a crea arhiva ZIP protejată cu parolă.

Cod:

```
if password_var.get():

    try:
```

```

password = password_entry.get()

if not password:

    tkinter.messagebox.showwarning("Parolă lipsă", "Introduceți o parolă pentru arhivă!")

    return

pyminizip.compress_multiple(file_list, [], zip_path, password, compression_method)

    tkinter.messagebox.showinfo("Succes", f"Arhivarea cu parolă s-a efectuat cu succes la {zip_path}")

except Exception as e:

    tkinter.messagebox.showerror("Eroare", f"A apărut o eroare: {e}")

```

Configurarea interfeței grafice

Interfața grafică include mai multe elemente: un titlu principal, butoane pentru selectarea fișierelor și folderelor, opțiuni pentru nivelul de compresie și adăugarea parolei, precum și un buton pentru crearea arhivei ZIP.

Cod:

```

file_list = []

# Titlu principal

title_label = ctk.CTkLabel(root, text="Creare fișier ZIP", font=ctk.CTkFont(size=24, weight="bold"))

title_label.pack(pady=20)

# Frame scrollabil pentru lista fișierelor

file_frame = ctk.CTkScrollableFrame(root, width=500, height=200)

file_frame.pack(pady=10)

# Butoane pentru selectarea fișierelor și folderelor

select_files_button = ctk.CTkButton(root, text="Selectează fișiere", command=select_files, width=200)

select_files_button.pack(pady=5)

select_folder_button = ctk.CTkButton(root, text="Selectează folder", command=select_folder, width=200)

select_folder_button.pack(pady=5)

# Opțiunea de selectare a nivelului de compresie

```

```

compression_label = ctk.CTkLabel(root, text="Nivel de compresie:", font=ctk.CTkFont(size=14))

compression_label.pack(pady=10)

compression_level = ctk.StringVar(value="Fără compresie (rapid)")

compression_dropdown = ctk.CTkComboBox(root, values=["Fără compresie (rapid)", "Compresie maximă"], variable=compression_level, state="readonly", width=300)

compression_dropdown.pack(pady=5)

# Opțiunea de parolă

password_var = ctk.BooleanVar(value=False)

password_checkbox = ctk.CTkCheckBox(root, text="Adaugă parolă", variable=password_var)

password_checkbox.pack(pady=10)

password_entry = ctk.CTkEntry(root, placeholder_text="Introduceți parola", show="*")

password_entry.pack(pady=5)

# Buton pentru crearea arhivei ZIP

zip_button = ctk.CTkButton(root, text="Creează ZIP", command=create_zip, width=200, fg_color="green")

zip_button.pack(pady=20)

```

Instrucțiuni de utilizare

- Selectează fișiere:** Apasă butonul „Selectează fișiere” pentru a alege fișierele care vor fi adăugate în arhivă.
- Selectează un folder:** Apasă „Selectează folder” pentru a adăuga toate fișierele dintr-un folder.
- Alege nivelul de compresie:** Selectează opțiunea dorită din dropdown-ul pentru nivelul de compresie.
- Adaugă parolă:** Bifează opțiunea „Adaugă parolă” și introdu parola dorită.
- Crează arhiva ZIP:** Apasă butonul „Creează ZIP” pentru a crea arhiva ZIP.

Crearea executabilului

Pentru a crea un executabil din acest cod Python folosind PyInstaller, folosește următoarea comandă:

```
pyinstaller --onefile --add-data "nume_icon.ico;." --icon=nume_icon.ico  
numefisier.py
```

Instalarea dependențelor

1. Asigură-te că ai instalat Python 3.x.
2. Instalează dependențele necesare:
`pip install customtkinter pyminizip`
3. Instalează Pillow pentru gestionarea imaginilor:
`pip install Pillow`