# Font Library Application Codebase Documentation

## Introduction

The Font Library is a user-friendly desktop application designed to simplify the process of

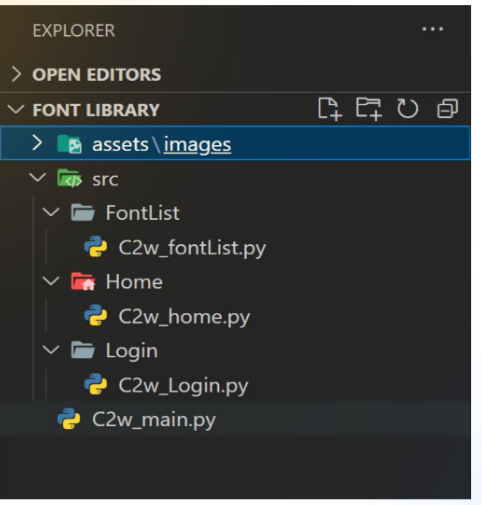
selecting and previewing fonts for text input. It offers a streamlined interface that allows users to

effortlessly experiment with different font styles before making a final selection.

# Title

Font Library Application

# Project Folder Structure



Python

Python

Code

1. C2w\_main.py

from Login.C2w\_Login import MainFrame

if \_\_name\_\_ == "\_\_main\_\_":

app = MainFrame(None, None, None)

app.run()

2. src/Login/C2w\_Login.py

import tkinter as tk

from PIL import ImageTk, Image

from google.cloud import firestore

import json

from tkinter import messagebox

class MainFrame():

def \_\_init\_\_(self, parent, main\_frame, userData):

self.useData = userData

self.root = tk.Tk()

self.root.geometry('1400x800')

self.root.resizable(0, 0)

self.root.title("Font Library by Core2web")

self.root.iconbitmap("./assets/images/core2web.png")

self.logIn()

def logIn(self):

# Create a frame to hold the UI

self.main\_frame = tk.Frame(self.root, height=700 , width=100)

self.main\_frame.pack(fill=tk.BOTH, expand=True)

#

========================================================================

self.bg\_frame = Image.open('./assets/images/background1.png')

photo = ImageTk.PhotoImage(self.bg\_frame)

self.bg\_panel = tk.Label( self.main\_frame, image=photo)

self.bg\_panel.image = photo

self.bg\_panel.pack(fill='both', expand='yes')

# ====== Login Frame =========================

self.lgn\_frame = tk.Frame(self.main\_frame, bg='#040405', width=990,

height=650)

self.lgn\_frame.place(x=200, y=70)

#

========================================================================

self.txt = "WELCOME"

self.txt = "CORE2WEB"

self.heading = tk.Label(self.lgn\_frame, text=self.txt, font=('yu gothic

ui', 25, "bold"), bg="#040405",fg='white',bd=5,relief=tk.FLAT)

self.heading.place(x=80, y=30, width=300, height=30)

#

========================================================================

# ============ Left Side Image

================================================

#

========================================================================

self.side\_image = Image.open('./assets/images/core2web.png')

photo = ImageTk.PhotoImage(self.side\_image)

self.side\_image\_label = tk.Label(self.lgn\_frame, image=photo,

bg='#040405')

self.side\_image\_label.image = photo

self.side\_image\_label.place(x=5, y=100)

#

========================================================================

# ============ Sign In Image

=============================================

#

========================================================================

self.sign\_in\_image = Image.open('./assets/images/hyy.png')

photo = ImageTk.PhotoImage(self.sign\_in\_image)

self.sign\_in\_image\_label = tk.Label(self.lgn\_frame, image=photo,

bg='#040405')

self.sign\_in\_image\_label.image = photo

self.sign\_in\_image\_label.place(x=620, y=130)

#

========================================================================

# ============ Sign In label

=============================================

#

========================================================================

self.sign\_in\_label = tk.Label(self.lgn\_frame, text="Sign In",

bg="#040405", fg="white",font=("yu gothic ui", 17, "bold"))

self.sign\_in\_label.place(x=650, y=240)

#

========================================================================

#

============================username====================================

#

========================================================================

self.username\_label = tk.Label(self.lgn\_frame, text="Username",

bg="#040405", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))

self.username\_label.place(x=550, y=300)

self.username\_entry = tk.Entry(self.lgn\_frame, highlightthickness=0,

relief=tk.FLAT, bg="#040405", fg="#6b6a69",font=("yu gothic ui ", 12, "bold"),

insertbackground = '#6b6a69')

self.username\_entry.place(x=580, y=335, width=270)

self.username\_line = tk.Canvas(self.lgn\_frame, width=300, height=2.0,

bg="#bdb9b1", highlightthickness=0)

self.username\_line.place(x=550, y=359)

# ===== Username icon =========

self.username\_icon = Image.open('./assets/images/username\_icon.png')

photo = ImageTk.PhotoImage(self.username\_icon)

self.username\_icon\_label = tk.Label(self.lgn\_frame, image=photo,

bg='#040405')

self.username\_icon\_label.image = photo

self.username\_icon\_label.place(x=550, y=332)

# ============================login

button================================

self.lgn\_button = Image.open('./assets/images/btn1.png')

photo = ImageTk.PhotoImage(self.lgn\_button)

self.lgn\_button\_label = tk.Label(self.lgn\_frame, image=photo,

bg='#040405')

self.lgn\_button\_label.image = photo

self.lgn\_button\_label.place(x=550, y=450)

self.login = tk.Button(self.lgn\_button\_label, text='LOGIN', font=("yu

gothic ui", 13, "bold"), width=25, bd=0,bg='#3047ff', cursor='hand2',

activebackground='#3047ff', fg='white', command=self.switch\_to\_homeLabel)

self.login.place(x=20, y=10)

#

============================password====================================

self.password\_label = tk.Label(self.lgn\_frame, text="Password",

bg="#040405", fg="#4f4e4d",font=("yu gothic ui", 13, "bold"))

self.password\_label.place(x=550, y=380)

self.password\_entry = tk.Entry(self.lgn\_frame, highlightthickness=0,

relief=tk.FLAT, bg="#040405", fg="#6b6a69",font=("yu gothic ui", 12, "bold"),

show="\*", insertbackground = '#6b6a69')

self.password\_entry.place(x=580, y=416, width=244)

self.password\_line = tk.Canvas(self.lgn\_frame, width=300, height=2.0,

bg="#bdb9b1", highlightthickness=0)

self.password\_line.place(x=550, y=440)

# ======== Password icon ================

self.password\_icon = Image.open('./assets/images/password\_icon.png')

photo = ImageTk.PhotoImage(self.password\_icon)

self.password\_icon\_label = tk.Label(self.lgn\_frame, image=photo,

bg='#040405')

self.password\_icon\_label.image = photo

self.password\_icon\_label.place(x=550, y=414)

# ========= show/hide password

==================================================================

self.show\_image = ImageTk.PhotoImage \

(file='./assets/images/show.png')

self.hide\_image = ImageTk.PhotoImage \

(file='./assets/images/hide.png')

self.show\_button = tk.Button(self.lgn\_frame, image=self.show\_image,

command=self.show, relief=tk.FLAT,activebackground="white",borderwidth=0,

background="white", cursor="hand2")

self.show\_button.place(x=860, y=420)

def show(self):

self.hide\_button = tk.Button(self.lgn\_frame, image=self.hide\_image,

command=self.hide, relief=tk.FLAT,activebackground="white", borderwidth=0,

background="white", cursor="hand2")

self.hide\_button.place(x=860, y=420)

self.password\_entry.config(show='')

def hide(self):

self.show\_button = tk.Button(self.lgn\_frame, image=self.show\_image,

command=self.show, relief=tk.FLAT,activebackground="white", borderwidth=0,

background="white", cursor="hand2")

self.show\_button.place(x=860, y=420)

self.password\_entry.config(show='\*')

def switch\_to\_homeLabel(self):

from Home.C2w\_home import homeLabel

username\_value = self.username\_entry.get()

userpassword\_value = self.password\_entry.get()

if(username\_value == 'Core2web' and userpassword\_value == 'C2w@123'):

# Destroy the current frame

messagebox.showinfo("Succesfull", "Login sucessfully")

self.main\_frame.destroy()

# Create and display the new frame from Class1

class1\_instance = homeLabel(self.root, self, None)

class1\_instance.pack(fill=tk.BOTH, expand=True)

return

elif(username\_value != 'Core2web'):

messagebox.showinfo("Error", "Incorrect Username")

elif(userpassword\_value == 'C2w@123'):

messagebox.showinfo("Error", "Incorrect Password")

def run(self):

self.root.mainloop()

Python

3. src/Home/C2w\_home.py

import tkinter as tk

from tkinter import ttk

from tkinter import ttk, messagebox

import os

from PIL import Image, ImageTk

class homeLabel(tk.Frame):

def \_\_init\_\_(self, parent, main\_frame, userData):

super().\_\_init\_\_(parent)

self.parent = parent

self.main\_frame = main\_frame

self.configure(bg="lightblue")

self.userData = userData

# Add widgets to the new frame

self.c2w\_background\_frame = tk.Frame(self, width=1000, height=1000)

self.c2w\_background\_frame.pack(anchor=tk.CENTER, expand=True)

# Create the left frame for navigation and buttons

self.c2w\_left\_frame = tk.Frame(self.c2w\_background\_frame, bg="#00072D",

width=300, height=1000)

self.c2w\_left\_frame.pack(side=tk.LEFT, expand=True)

# Set up button styling

style = ttk.Style()

style.configure("Rounded.TButton", borderwidth=0, relief="flat",

background="#BCD2E8", padding=10, font=("Poppins", 12))

style.map("Rounded.TButton", foreground=[('pressed', 'black'),

('active', 'white')])

# Create the Font Library button

self.c2w\_button\_label = ttk.Button(self.c2w\_left\_frame, text="Font

Library", style="Rounded.TButton", command=self.c2w\_show\_font\_tab)

self.c2w\_button\_label.pack()

self.c2w\_button\_label.place(x=60, y=350, anchor='nw')

# Create the Font Library button

self.c2w\_button\_label = ttk.Button(self.c2w\_left\_frame, text="Font

List", style="Rounded.TButton", command=self.switch\_to\_class2)

self.c2w\_button\_label.pack()

self.c2w\_button\_label.place(x=60, y=450, anchor='nw')

# Load and display an image

script\_directory = os.path.dirname(os.path.abspath(\_\_file\_\_))

img\_path = os.path.join(script\_directory,

"../../assets/images/c2w.png")

img = ImageTk.PhotoImage(Image.open(img\_path))

self.c2w\_label = tk.Label(self.c2w\_left\_frame, image=img,

background="#00072D")

self.c2w\_label.image = img

self.c2w\_label.pack()

self.c2w\_label.place(x=40, y=20, anchor='nw')

# Set up the right frame for dynamic content

rgba\_color = (222, 55, 48, 255)

tk\_color = "#{:02x}{:02x}{:02x}".format(\*rgba\_color[:3])

self.c2w\_right\_frame = tk.Frame(self.c2w\_background\_frame,

bg="#BCD2E8", width=1200, height=1000)

self.c2w\_right\_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

# Set up a label with welcome text

self.c2w\_original\_label\_text = "Welcome To Core2web"

self.c2w\_right\_label = tk.Label(self.c2w\_right\_frame,

text=self.c2w\_original\_label\_text, background="#BCD2E8", font=("Poppins", 50,

"bold"), fg=tk\_color)

self.c2w\_right\_label.pack()

self.c2w\_right\_label.place(x=200, y=300, anchor='nw')

self.font\_library\_opened = False

self.additional\_widgets = []

def c2w\_show\_font\_tab(self):

# Show Font Library tab

if self.font\_library\_opened:

messagebox.showinfo("Already Opened", "Font Library is already

opened!")

else:

# Destroy existing widgets

for widget in self.additional\_widgets:

widget.destroy()

self.additional\_widgets = []

# Set up Font Library tab

rgba\_color = (222, 55, 48, 255)

tk\_color = "#{:02x}{:02x}{:02x}".format(\*rgba\_color[:3])

# Create and pack widgets for Font Library

self.c2w\_heading\_label = tk.Label(self.c2w\_right\_frame, text="Font

Library", background="#BCD2E8", font=("Poppins", 20, "bold"), fg=tk\_color)

self.c2w\_heading\_label.pack()

self.c2w\_heading\_label.place(x=60\*2, y=150, anchor='nw')

self.input\_label = ttk.Entry(self.c2w\_right\_frame, width=60,

font=("Poppins", 12))

self.input\_label.insert(0, "")

self.input\_label.pack()

self.input\_label.place(x=60\*2, y=200, anchor='nw')

self.c2w\_right\_label.config(text="")

self.font\_library\_opened = True

# Create buttons for different fonts

self.c2w\_button\_label1 = ttk.Button(self.c2w\_right\_frame,

text="Poppins", style="Rounded.TButton", command=self.c2w\_poppinClicked)

self.c2w\_button\_label1.pack()

self.c2w\_button\_label1.place(x=60\*2, y=350, anchor='nw')

self.c2w\_button\_label2 = ttk.Button(self.c2w\_right\_frame,

text="Modern", style="Rounded.TButton", command=self.c2w\_modernClicked)

self.c2w\_button\_label2.pack()

self.c2w\_button\_label2.place(x=60\*5, y=350, anchor='nw')

self.c2w\_button\_label3 = ttk.Button(self.c2w\_right\_frame,

text="Courier", style="Rounded.TButton", command=self.c2w\_courierClicked)

self.c2w\_button\_label3.pack()

self.c2w\_button\_label3.place(x=60\*8, y=350, anchor='nw')

self.c2w\_button\_label4 = ttk.Button(self.c2w\_right\_frame, text="MS

Sans Serif", style="Rounded.TButton", command=self.c2w\_sanserifClicked)

self.c2w\_button\_label4.pack()

self.c2w\_button\_label4.place(x=60\*11, y=350, anchor='nw')

self.c2w\_button\_label5 = ttk.Button(self.c2w\_right\_frame,

text="Calibri", style="Rounded.TButton", command=self.c2w\_calibriClicked)

self.c2w\_button\_label5.pack()

self.c2w\_button\_label5.place(x=60\*14, y=350, anchor='nw')

self.c2w\_show\_label = tk.Label(self.c2w\_right\_frame, text="Hello

from Core2web", background="#BCD2E9", font=("Poppins", 50, "bold"),

fg=tk\_color)

self.c2w\_show\_label.pack()

self.c2w\_show\_label.place(x=60\*2, y=500, anchor='nw')

def c2w\_poppinClicked(self):

# Change the font and display entered text

self.c2w\_show\_label.config(text=self.input\_label.get(),

font=("Poppins", 30, "bold"))

def c2w\_modernClicked(self):

# Change the font and display entered text

self.c2w\_show\_label.config(text=self.input\_label.get(), font=("Modern",

30, "bold"))

def c2w\_courierClicked(self):

# Change the font and display entered text

self.c2w\_show\_label.config(text=self.input\_label.get(),

font=("Courier", 30, "bold"))

def c2w\_sanserifClicked(self):

# Change the font and display entered text

self.c2w\_show\_label.config(text=self.input\_label.get(), font=("MS Sans

Serif", 30, "bold"))

def c2w\_calibriClicked(self):

# Change the font and display entered text

self.c2w\_show\_label.config(text=self.input\_label.get(),

font=("Calibri", 30, "bold"))

def switch\_to\_class2(self):

from FontList.C2w\_fontList import FontListApp

# Destroy the current frame

self.destroy()

# Create and display the new frame from Class2

class2\_instance = FontListApp(self.parent, self.main\_frame,

self.userData)

class2\_instance.pack(fill=tk.BOTH, expand=True)

def c2w\_reset\_ui(self):

Python

# Reset UI when going back

self.font\_library\_opened = False

for widget in self.additional\_widgets:

widget.destroy()

self.additional\_widgets = []

self.c2w\_heading\_label.destroy()

self.input\_label.destroy()

self.c2w\_button\_label1.destroy()

self.c2w\_button\_label2.destroy()

self.c2w\_button\_label3.destroy()

self.c2w\_button\_label4.destroy()

self.c2w\_button\_label5.destroy()

self.c2w\_show\_label.destroy()

self.c2w\_right\_label.config(text=self.c2w\_original\_label\_text)

4. src/FontList/C2w\_fontList.py

import tkinter as tk

from tkinter import ttk

from tkinter import font

class FontListApp(tk.Frame):

def \_\_init\_\_(self, parent, main\_frame, userData):

super().\_\_init\_\_(parent)

self.parent = parent

self.main\_frame = main\_frame

self.configure(bg="#BCD2E8")

self.userData = userData

# Get the list of available fonts

available\_fonts = font.families()

# Create the main background frame

self.c2w\_background\_frame = tk.Frame(self, bg="#BCD2E8", width=800,

height=600)

self.c2w\_background\_frame.pack(fill=tk.BOTH, expand=True)

# Create a text widget to display the font list

font\_list\_text = tk.Text(self.c2w\_background\_frame, wrap=tk.WORD,

height=20, width=40, bg="white", fg="black", font=("Helvetica", 12))

font\_list\_text.pack(padx=50, pady=20, fill=tk.BOTH, expand=True)

# Insert the font list into the text widget

# Insert each font name into the text widget with index

for i, font\_family in enumerate(available\_fonts):

bg\_color = "#FFFFFF" if i % 2 == 0 else "#EFEFEF"

font\_list\_text.insert(tk.END, f"{i + 1}. {font\_family}\n",

bg\_color)

# Style the text widget

font\_list\_text.config(state=tk.DISABLED) # Disable editing

font\_list\_text.config(cursor="arrow") # Change cursor to arrow

# Create a button to switch back to the home label

switch\_button = ttk.Button(self, text="Back",

command=self.switch\_to\_class1, style="Rounded.TButton")

switch\_button.pack(pady=20)

def switch\_to\_class1(self):

from Home.C2w\_home import homeLabel

# Destroy the current frame

self.destroy()

# Create and display the new frame from Class1

class1\_instance = homeLabel(self.parent, self.main\_frame,

self.userData)

class1\_instance.pack(fill=tk.BOTH, expand=True)

def apply\_styles():

# Create a custom style for rounded buttons

style = ttk.Style()

style.configure("Rounded.TButton",

foreground="white",

background="#3047ff",

font=("Helvetica", 12),

relief=tk.FLAT)

style.map("Rounded.TButton",

foreground=[("active", "white"), ("pressed", "white")],

background=[("active", "#3047ff"), ("pressed", "#3047ff")])