```
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import qrcode
from PIL import Image, ImageTk
import os

class QRGenerator:
    def __init__(self, root):
        self.root = root
        self.root.title("Super QR Generator")
        self.root.geometry("500x650")
        self.root.configure(bg="#2C3E50")

        # Custom Styling
        self.style = ttk.Style()
        self.style.configure("TButton", font=("Helvetica", 12), padding=10)
        self.style.configure("TLabel", background="#2C3E50", foreground="white")

        # Header
        self.header = ttk.Label(root, text="🎨 Smart QR Generator",
font=("Helvetica", 20, "bold"))
        self.header.pack(pady=20)

        # Input Section
        self.input_frame = ttk.Frame(root)
        self.input_frame.pack(pady=10)

        self.data_type = tk.StringVar(value="url")

        ttk.Radiobutton(self.input_frame, text="Phone", variable=self.data_type,
value="phone", style="TLabel").grid(row=0, column=0, padx=5)
        ttk.Radiobutton(self.input_frame, text="URL/Link", variable=self.data_type,
value="url", style="TLabel").grid(row=0, column=1, padx=5)
        ttk.Radiobutton(self.input_frame, text="PDF", variable=self.data_type,
value="pdf", style="TLabel").grid(row=0, column=2, padx=5)

        self.entry = ttk.Entry(root, width=40, font=("Helvetica", 12))
        self.entry.pack(pady=20, ipady=5)

        # Generate Button
        self.gen_btn = ttk.Button(root, text="Generate QR Code",
command=self.generate_qr)
        self.gen_btn.pack()

        # QR Display
        self.qr_label = ttk.Label(root, background="white")
        self.qr_label.pack(pady=20)

        # Save Button
        self.save_btn = ttk.Button(root, text="Save QR Code", command=self.save_qr,
state=tk.DISABLED)
        self.save_btn.pack(pady=10)

    def generate_qr(self):
        data = self.entry.get()
        if not data:
            messagebox.showerror("Error", "Please enter some data!")
            return
        try:
            qr = qrcode.QRCode(
                version=1,
                error_correction=qrcode.constants.ERROR_CORRECT_H,
                box_size=10,
                border=4,
            )

            # Add proper prefix based on data type
            if self.data_type.get() == "phone":
                data = f"tel:{data}"
            elif self.data_type.get() == "pdf":
                if not data.lower().endswith(('.pdf', '/')):
                    data += ".pdf"

            qr.add_data(data)
            qr.make(fit=True)

            img = qr.make_image(fill_color="#E74C3C", 
back_color="white").convert('RGB')
            img = img.resize((250, 250), Image.LANCZOS)

            self.tk_image = ImageTk.PhotoImage(img)
            self.qr_label.configure(image=self.tk_image)
            self.save_btn.config(state=tk.NORMAL)

        except Exception as e:
            messagebox.showerror("Error", f"Generation failed: {str(e)}")

    def save_qr(self):
        path = filedialog.asksaveasfilename(
            defaultextension=".png",
            filetypes=[("PNG files", "*.png"), ("All files", "*.*")]
        )
        if path:
            try:
                self.tk_image._PhotoImage__photo.write(path, format='png')
                messagebox.showinfo("Success", f"QR Code saved successfully at:
\n{path}")
            except Exception as e:
                messagebox.showerror("Error", f"Save failed: {str(e)}")

if __name__ == "__main__":
    root = tk.Tk()
    app = QRGenerator(root)
    root.mainloop()
```