```
import tkinter as tk
from tkinter import messagebox, filedialog
import sqlite3
import qrcode
import barcode
from barcode.writer import ImageWriter
from PIL import Image, ImageDraw, ImageFont
import os

# Database setup
conn = sqlite3.connect("inventory.db")
cursor = conn.cursor()
cursor.execute("""
CREATE TABLE IF NOT EXISTS products (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    price REAL NOT NULL,
    code TEXT NOT NULL UNIQUE
)
""")
conn.commit()

# Generate barcode image
def generate_barcode(code, path):
    ean = barcode.get('code128', code, writer=ImageWriter())
    ean.save(path)

# Generate QR code image
def generate_qrcode(data, path):
    img = qrcode.make(data)
    img.save(path)

# Export label as image
def export_label(name, price, code):
    barcode_path = "barcode.png"
    qrcode_path = "qrcode.png"
    generate_barcode(code, "barcode")
    generate_qrcode(f"{name} - ${price}", "qrcode.png")

    # Create label canvas
    label = Image.new('RGB', (400, 200), 'white')
    draw = ImageDraw.Draw(label)
    font = ImageFont.load_default()
    draw.text((10, 10), f"Name: {name}", fill='black', font=font)
    draw.text((10, 30), f"Price: ${price}", fill='black', font=font)
    draw.text((10, 50), f"Code: {code}", fill='black', font=font)

    # Paste barcode and QR code
    bc_img = Image.open(barcode_path).resize((180, 60))
    qr_img = Image.open(qrcode_path).resize((100, 100))
    label.paste(bc_img, (10, 80))
    label.paste(qr_img, (250, 50))

    # Save final label
     output_path = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG
```

```
files", "*.png")])
    if output_path:
        label.save(output_path)
        messagebox.showinfo("Success", f"Label saved to {output_path}")

# Save product data
def save_product():
    name = name_var.get()
    price = price_var.get()
    code = code_var.get()

    if not name or not price or not code:
        messagebox.showerror("Error", "All fields are required.")
        return

    try:
            cursor.execute("INSERT INTO products (name, price, code) VALUES (?, ?, ?)",
(name, float(price), code))
        conn.commit()
        export_label(name, float(price), code)
        messagebox.showinfo("Saved", "Product saved and label exported.")
    except sqlite3.IntegrityError:
        messagebox.showerror("Error", "Code must be unique.")
    except ValueError:
        messagebox.showerror("Error", "Invalid price format.")

# GUI setup
root = tk.Tk()
root.title("Inventory Label Tool")

tk.Label(root, text="Product Name").grid(row=0, column=0, padx=10, pady=5)
tk.Label(root, text="Price").grid(row=1, column=0, padx=10, pady=5)
tk.Label(root, text="Code").grid(row=2, column=0, padx=10, pady=5)

name_var = tk.StringVar()
price_var = tk.StringVar()
code_var = tk.StringVar()

tk.Entry(root, textvariable=name_var).grid(row=0, column=1, padx=10, pady=5)
tk.Entry(root, textvariable=price_var).grid(row=1, column=1, padx=10, pady=5)
tk.Entry(root, textvariable=code_var).grid(row=2, column=1, padx=10, pady=5)

tk.Button(root, text="Save and Export Label", command=save_product).grid(row=3,
columnspan=2, pady=15)

root.mainloop()
```