MODERN IMAGE VIEWER

```
import os

import tkinter as tk

from tkinter import filedialog, messagebox

from PIL import Image, ImageTk


class ImageViewer:

    def _init_(self, root):

        self.root = root

        self.root.title("🖼️ Modern Image Viewer")

        self.root.geometry("1000x600")

        self.root.configure(bg="#2c3e50")


        self.image_list = []

        self.current_image = 0

        self.zoom_level = 1.0

        self.slideshow_active = False

        self.fullscreen = False


        try:

            self.resample_method = Image.Resampling.LANCZOS

        except AttributeError:

            self.resample_method = Image.ANTIALIAS


        self.label = tk.Label(
```

```
    self.root, text="Image Viewer", font=("Helvetica", 16, "bold"),
    bg="#34495e", fg="#ecf0f1", pady=10
)
self.label.pack(fill=tk.X)

    self.canvas = tk.Canvas(self.root, bg="black", width=900, height=500,
highlightthickness=0)
    self.canvas.pack(pady=10)


    btn_frame = tk.Frame(self.root, bg="#2c3e50")
    btn_frame.pack(pady=10)


    btn_style = {"padx": 15, "pady": 5, "bg": "#2980b9", "fg": "white", "font":
("Arial", 10, "bold")}

    tk.Button(btn_frame, text="Open Folder", command=self.open_folder,
**btn_style).grid(row=0, column=0, padx=5)
    tk.Button(btn_frame, text="Previous", command=self.prev_image,
**btn_style).grid(row=0, column=1, padx=5)
    tk.Button(btn_frame, text="Next", command=self.next_image,
**btn_style).grid(row=0, column=2, padx=5)
    tk.Button(btn_frame, text="Zoom In", command=self.zoom_in,
**btn_style).grid(row=0, column=3, padx=5)
    tk.Button(btn_frame, text="Zoom Out", command=self.zoom_out,
**btn_style).grid(row=0, column=4, padx=5)
    tk.Button(btn_frame, text="Start Slide", command=self.start_slideshow,
**btn_style).grid(row=0, column=5, padx=5)
    tk.Button(btn_frame, text="Stop Slide", command=self.stop_slideshow,
**btn_style).grid(row=0, column=6, padx=5)
```

```
        tk.Button(btn_frame, text="Fullscreen", command=self.toggle_fullscreen,
**btn_style).grid(row=0, column=7, padx=5)

        tk.Button(btn_frame, text="Exit", command=self.exit_app,
**btn_style).grid(row=0, column=8, padx=5)

    # Bind ESC key to exit fullscreen
    self.root.bind("<Escape>", self.exit_fullscreen)


def open_folder(self):
    folder_selected = filedialog.askdirectory()
    if folder_selected:
        supported_formats = (".jpg", ".jpeg", ".png", ".bmp", ".gif")
        self.image_list = [
            os.path.join(folder_selected, f)
    for f in os.listdir(folder_selected)
    if f.lower().endswith(supported_formats)
]
self.image_list.sort()

if self.image_list:
    self.current_image = 0
    self.zoom_level = 1.0
            self.display_image()
        else:
            messagebox.showwarning("No Images", "No supported image files
found in this folder.")
```

```
def display_image(self):
    try:
        image_path = self.image_list[self.current_image]
        img = Image.open(image_path)

        canvas_width = self.canvas.winfo_width()
        canvas_height = self.canvas.winfo_height()


        img_width, img_height = img.size
        scale = min((canvas_width / img_width), (canvas_height / img_height))
* self.zoom_level
        new_width = int(img_width * scale)
        new_height = int(img_height * scale)

        img = img.resize((new_width, new_height), self.resample_method)


        self.tk_img = ImageTk.PhotoImage(img)
        self.canvas.delete("all")
        self.canvas.create_image(canvas_width // 2, canvas_height // 2,
anchor=tk.CENTER, image=self.tk_img)


        self.label.config(text=os.path.basename(image_path))
    except Exception as e:
        messagebox.showerror("Error", f"Unable to load image.\n{str(e)}")


def next_image(self):
    if self.image_list:
```

```
            self.current_image = (self.current_image + 1) % len(self.image_list)
            self.zoom_level = 1.0
            self.display_image()

    def prev_image(self):
        if self.image_list:
            self.current_image = (self.current_image - 1) % len(self.image_list)
            self.zoom_level = 1.0
            self.display_image()

    def zoom_in(self):
        self.zoom_level *= 1.25
        self.display_image()

    def zoom_out(self):
        self.zoom_level /= 1.25
        self.display_image()

    def start_slideshow(self):
        if not self.slideshow_active:
            self.slideshow_active = True
            self.run_slideshow()

    def stop_slideshow(self):
        self.slideshow_active = False
```

```
def run_slideshow(self):
    if self.slideshow_active and self.image_list:
        self.next_image()
        self.root.after(2000, self.run_slideshow)

def toggle_fullscreen(self):
    self.fullscreen = not self.fullscreen
    self.root.attributes("-fullscreen", self.fullscreen)
    if self.fullscreen:
        self.label.config(text="Fullscreen Mode (Press ESC to exit)")


def exit_fullscreen(self, event=None):
    """Exit fullscreen mode"""
    if self.fullscreen:
        self.fullscreen = False
        self.root.attributes("-fullscreen", False)
        self.label.config(text="Image Viewer")


def exit_app(self):
    """Exit the entire application"""
    self.stop_slideshow()
        self.root.destroy()  # safer than quit()


if _name_ == "_main_":
    root = tk.Tk()
```

```
    app = ImageViewer(root)
    root.mainloop()
```

OUTPUT: