MACH IT

FINAL PROJECT WORK

IMAGE COLLAGE CREATION WITH OPEN CV EFFECTS

- LAKSHME NARAYANA R

1. Create a collage using open CV with minimum 8 images and add the effects like converting to Gray scale, pencil sketching, shadowing, frames etc.

DESCRIPTION:

This Project utilizes the OpenCV library to create a visually appealing image collage from a set of input images. The collage is enhanced with various effects and filters such as grayscale conversion, pencil sketching, shadowing, and frames.

- The sample Images and the code should be in same folder.
- After typing image title, press submit button.
- We can open single image, can add frame to it, can add pencil sketch edit to it and so on.
- For collage, Type Image name and submit one by one.
- For closing all the uploaded images, press reset button.

TECHNOLOGIES USED:

- Python
- OpenCV
- NumPy

CODE:

```
import cv2

import numpy as np

import tkinter as tk

def create_collage(images, rows, cols):

    max_height = max(image.shape[0] for image in images)
    total_width = sum(image.shape[1] for image in images)
    collage = np.zeros((max_height * rows, total_width // rows, 3),
dtype=np.uint8)
    x_offset, y_offset = 0, 0
    for image in images:
        height, width = image.shape[:2]
        collage[y_offset:y_offset+height, x_offset:x_offset+width] = image
        x_offset += width
        if x_offset >= total_width // rows:
            x_offset = 0
            y_offset += height
    return collage
def resetimages():
    print("Resetting....")
    images.clear()
    print("Resetted !!")
def on_entry_click(event):
    if entry.get() == 'Enter text here...':
        entry.delete(0, tk.END)
        entry.config(fg='black')
def on_focus_out(event):
    if not entry.get():
        entry.insert(0, 'Enter text here...')
        entry.config(fg='grey')
```

```
def submit_action():
    input_text = entry.get()
    txt=input_text+'.jpg'
    images.append(cv2.imread(txt))
    print("Submitted text:", txt)
def collage():
    global images
    if any(image is None for image in images):
        exit(0)
    else:
        height, width = 200, 200
        images = [cv2.resize(image, (width, height)) for image in images]
        if(len(images)%2!=0):
            images.pop(len(images)-1)
        collage = create_collage(images, rows=2, cols=len(images)/2)
        images.append(collage)
        cv2.imshow('Collage', collage)
        cv2.waitKey(0)
        cv2.destroyAllWindows()
def grayscale():
    image=images[len(images)-1]
    image = cv2.resize(image, (500, 300))
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    cv2.imshow('Grayscale Image', gray_image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
def add_shadow(image, intensity=0.5):
    shadow_overlay = np.zeros_like(image, dtype=np.uint8)
    shadow_color = (0, 0, 0)
    height, width = image.shape[:2]
    shadow_height = int(height * 0.2)
    shadow_mask = np.zeros((height, width), dtype=np.uint8)
    shadow_mask[height - shadow_height:, :] = 255
    shadow_overlay = cv2.merge([shadow_overlay[:, :, 0] + shadow_color[0],
                                shadow_overlay[:, :, 1] + shadow_color[1],
                                shadow_overlay[:, :, 2] + shadow_color[2]])
    shadowed_image = cv2.addWeighted(image, 1 - intensity, shadow_overlay, intensity, 0)
    return shadowed_image
def openimage():
    image=images[-1]
    image = cv2.resize(image, (500, 300))
    cv2.imshow('Resized Image',image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
def shadow():
    image = images[len(images)-1]
    image = cv2.resize(image, (500, 300))
    shadowed_image = add_shadow(image)
    cv2.imshow('Shadowed Image', shadowed_image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
def dodgeV2(image, mask):
    return cv2.divide(image, 255 - mask, scale=256)
def burnV2(image, mask):
    return 255 - cv2.divide(255 - image, 255 - mask, scale=256)
def pencil_sketch(image):
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    inverted_gray = 255 - gray_image
    blurred_image = cv2.GaussianBlur(inverted_gray, (21, 21), 0)
    pencil_sketch_image = dodgeV2(gray_image, blurred_image)
    pencil_sketch_image = burnV2(pencil_sketch_image, gray_image)
    return pencil_sketch_image
def sketch():
    image = images[len(images)-1]
    image = cv2.resize(image, (500, 300))
    pencil_sketch_image = pencil_sketch(image)
    cv2.imshow('Pencil Sketch', pencil_sketch_image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

```
def frameimage()
    image = images[len(images)-1]
    image = cv2.resize(image, (500, 300))
    frame_color = (100, 100, 100)
    thickness = 20
    height, width = image.shape[:2]
    image_with_frame = cv2.rectangle(image, (0, 0), (width-1, height-1), frame_color, thickness)
    cv2.imshow('Image with Frame', image_with_frame)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
images=[]
root = tk.Tk()
root.title("Frame with Title and Placeholder Example")
frame_width = 800
frame_height = 800
frame = tk.Frame(root, width=frame_width, height=frame_height, bg="lightblue")
frame.pack()
title_label = tk.Label(frame, text="PHOTO EDITOR", font=("Arial", 30), bg="lightblue")
title_label.place(relx=0.5, rely=0.1, anchor="center")
default_text = 'Filename'
entry = tk.Entry(frame, font=("Arial", 16), width=20, fg='grey')
entry.insert(0, default_text)
entry.place(relx=0.5, rely=0.4, anchor="center")

submit_button = tk.Button(frame, text="Submit", font=("Arial", 14), command=submit_action)

submit_button.place(relx=0.75, rely=0.4, anchor="center")

entry.bind('<Return>',submit_action)

but1 = tk.Button(frame, text="Gray Scale", font=("Arial", 14), command=grayscale)

but1.place(relx=0.35, rely=0.5, anchor="center")

but2 = tk.Button(frame, text="Shadow", font=("Arial", 14), command=shadow)

but2.place(relx=0.35, rely=0.6, anchor="center")

but3 = tk.Button(frame, text="Pencil Sketch", font=("Arial", 14), command=sketch)

but3.place(relx=0.35, rely=0.7, anchor="center")

but4 = tk.Button(frame, text="Add Frame", font=("Arial", 14), command=frameimage)

but4.place(relx=0.35, rely=0.8, anchor="center")

reset = tk.Button(frame, text="Reset", font=("Arial", 14), command=resetimages)

reset.place(relx=0.85, rely=0.4, anchor="center")

opener = tk.Button(frame, text="Open Image", font=("Arial", 14), command=openimage)

opener.place(relx=0.65, rely=0.5, anchor="center")

coll = tk.Button(frame, text="Make Collage", font=("Arial", 14), command=collage)

coll.place(relx=0.65, rely=0.6, anchor="center")

root.mainloop()
```

OUTPUT:

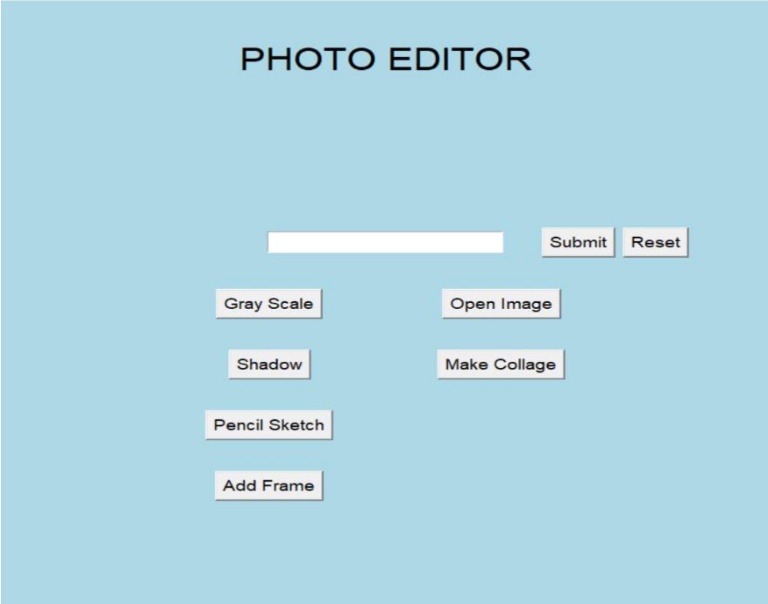
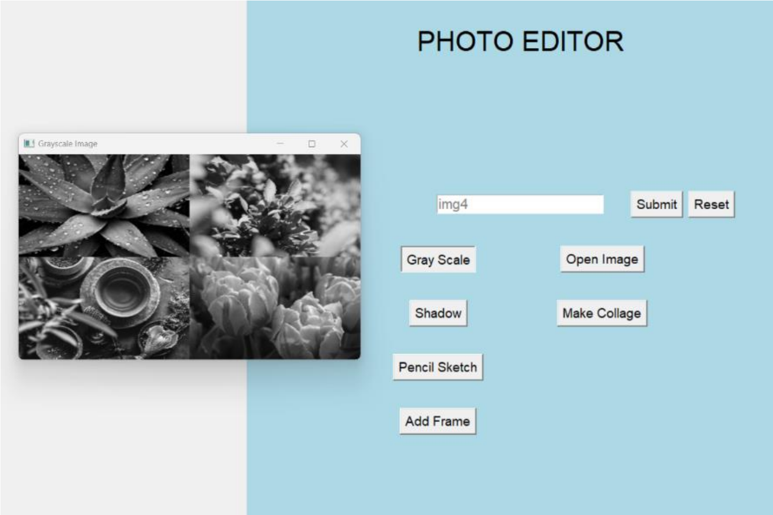
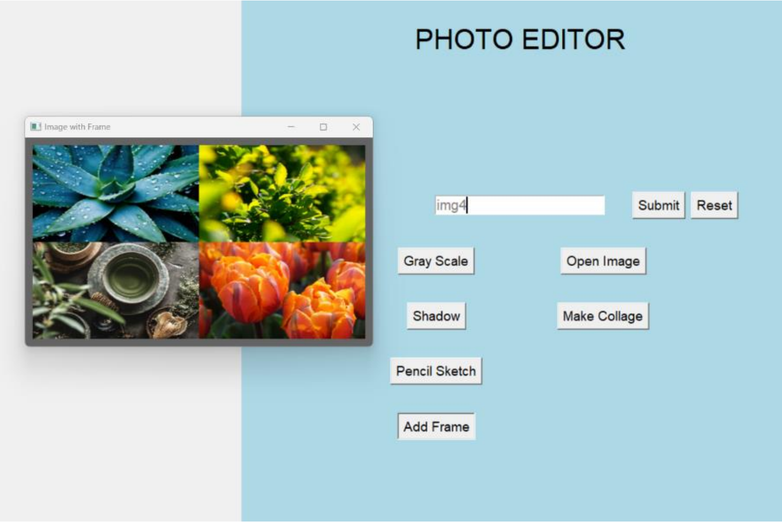
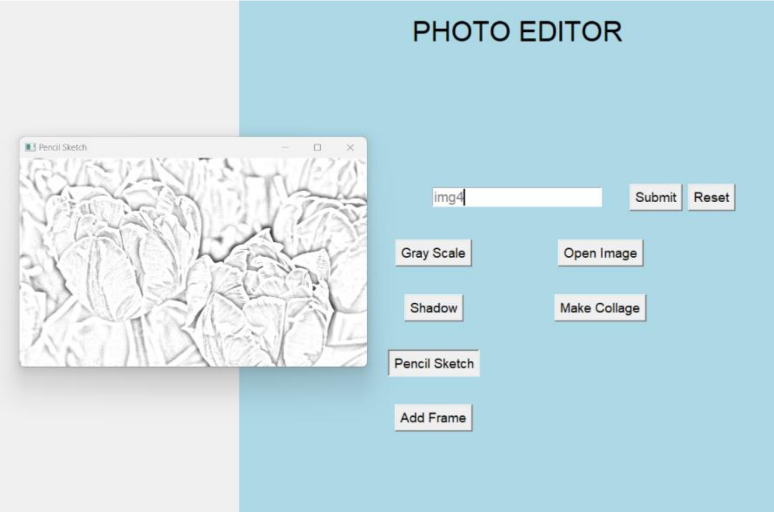


PHOTO EDITOR
Submit
Reset
Gray Scale
Open Image
Shadow
Make Collage
Pencil Sketch
Add Frame


Grayscale Image
PHOTO EDITOR
img4
Submit
Reset
Gray Scale
Open Image
Shadow
Make Collage
Pencil Sketch
Add Frame





RESULT:

This project demonstrates how to create a collage using OpenCV in Python and apply various effects like grayscale conversion, pencil sketching, shadowing, and frames to enhance the collage's visual appeal.