SYNOPSIS FORMAT FOR MICRO-PROJECT

1. **Group Members**
   3415 - Vibhavi Gaikwad
   3462 - Shuchi Talati
   3474 - Gayatri Walke

2. **Problem Statement/Title**
   Image Filters Creation

3. **Motivation**
   Zeal to create and explore different social media image filters (like Instagram and Snapchat, etc.)

4. **Keywords**
   OpenCV, Numpy, Tkinter, Filters, Laplacian

5. **Abstract(150 words approx.)**
   An image filter is a technique through which size, colors, shading and other characteristics of an image are altered. An image filter is used to transform the image using different graphical editing techniques. Our micro project creates the following filters namely, cartoon filter, pencil-sketch filter, invert filter using various functions from openCV.

6. **Technology Used**
   OpenCV, Numpy, Tkinter

7. **Literature Survey**
   OpenCV is a python library used for real-time computer vision applications. OpenCV is open source and has huge applications in Image Processing, Machine Learning and Deep Learning. OpenCV can be used for object detection, classification, handwriting analysis and much more.OpenCV can be integrated with libraries like Numpy and be used for various mathematical calculations.Many apps and websites provide tools to give an effect to our images. We just upload our image and an image with the desired effects are returned to us.To do this, multiple image transformations need to be done. Some important things

that need to be considered, while doing this are the edges and colour palette.

8. **Module wise Scope/Work flow/Block diagram with explanation**
   Import statements + opening an image
   Filtering
   Edge detection
   Color Quantization
   Combine the images

9. **Functions used**
   - GaussianBlur()
   - medianBlur()
   - bilateralFilter()
   - Laplacian()
   - cvtColor()
   - threshold()
   - subtract()
   - detailEnhance()
   - Bitwise_and()
   - imread()
   - imshow()

10. **References**
    - https://towardsdatascience.com/turn-photos-into-cartoons-using-python-bb1a9f578a7e

CODE :

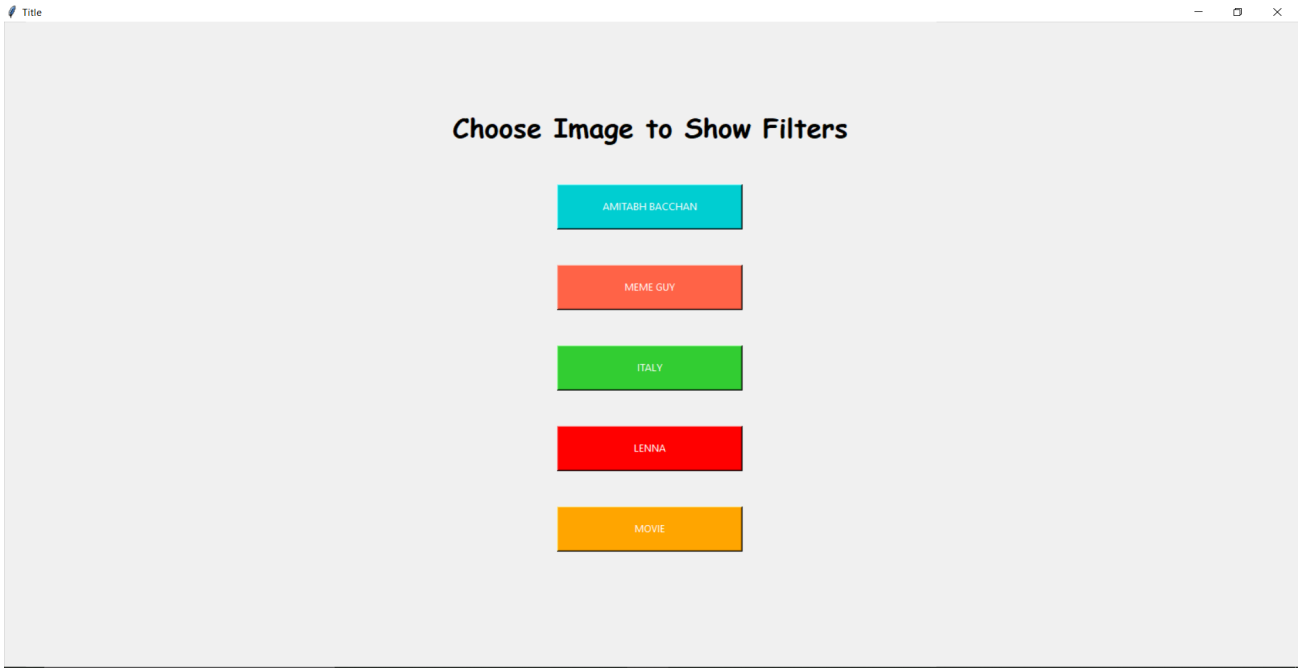
```
import cv2
import numpy as np
# from tkinter import *
import tkinter

root   tkinter.Tk()
```

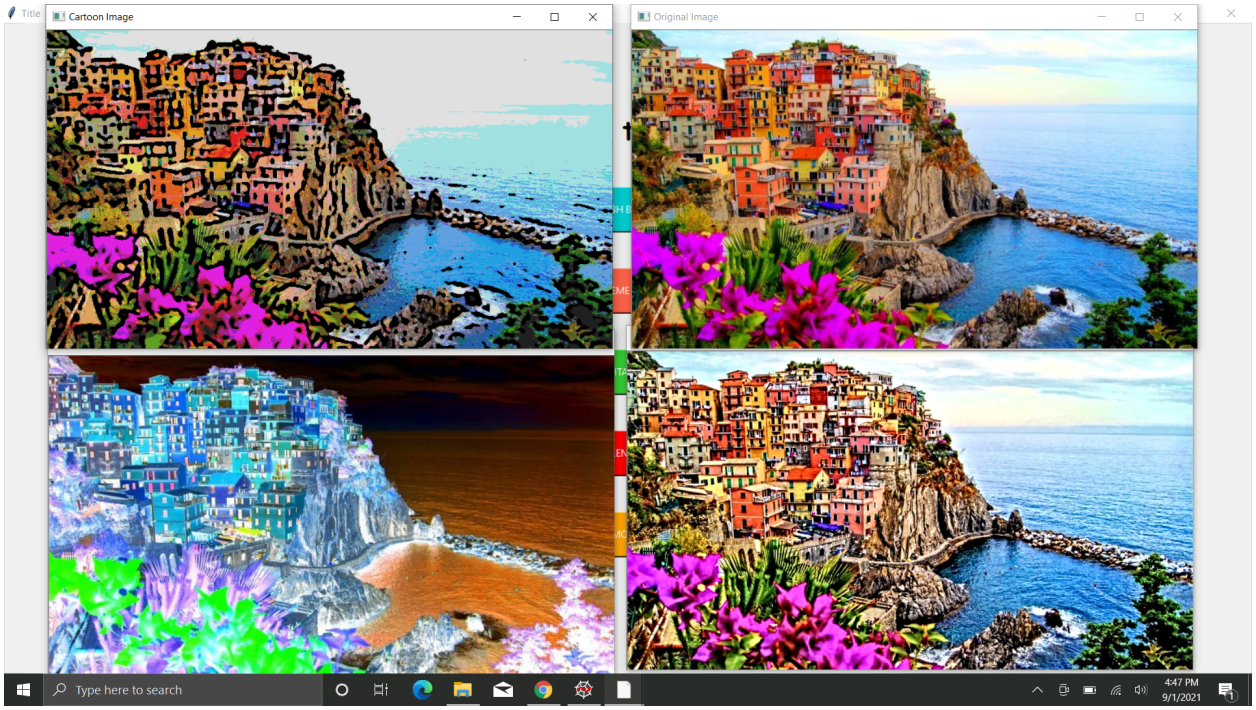
```
root.title("Title")
root.minsize(width 400, height 400)

canvas1   tkinter.Canvas(root, width 1000, height 1000)
canvas1.pack(padx 30, pady 30)

label0   tkinter.Label(root, text "Choose Image to Show Filters", font ("Comic
Sans MS", 24, "bold"))
canvas1.create_window(500, 100, window label0)


# my_label=tkinter.Label(text="My Label", font=("Comic Sans MS", 24, "bold"))
# my_label.pack()
def        (img)
   #Remove noise - apply filters

   # Apply some Gaussian blur on the image
   img_gb    cv2.GaussianBlur(img, (7, 7), 0)
   # Apply some Median blur on the image
   img_mb    cv2.medianBlur(img_gb, 5)
   # Apply a bilateral filer on the image
   '''
   d: Diameter of each pixel neighborhood.
   sigmaColor: Value of \sigma in the color space.
   The greater the value, the colors farther to each other will start to get
mixed.
   sigmaColor: Value of \sigma in the coordinate space. T
   he greater its value, the more further pixels will mix together, given that
their colors lie within the sigmaColor range.
   '''
   img_bf    cv2.bilateralFilter(img_mb, 5, 80, 80)

   # Use the laplace filter to detect edges
   img_lp_al    cv2.Laplacian(img_bf, cv2.CV_8U, ksize 5)

   #to get black and white image
   # Convert the image to greyscale (1D)
   img_lp_al_grey    cv2.cvtColor(img_lp_al, cv2.COLOR_BGR2GRAY)
   # Manual image thresholding
   _, EdgeImage    cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)


   # Remove some additional noise
   blur_al    cv2.GaussianBlur(img_lp_al_grey, (5, 5), 0)
   # Apply a threshold (Otsu)
   _, tresh_al    cv2.threshold(blur_al, 245, 255, cv2.THRESH_BINARY
cv2.THRESH_OTSU)
   # Invert the black and the white
   inverted_Bilateral    cv2.subtract(255, tresh_al)


   # Reduce the colors of the original image
   div    64
   img_bins    img    div    div    div    2
```

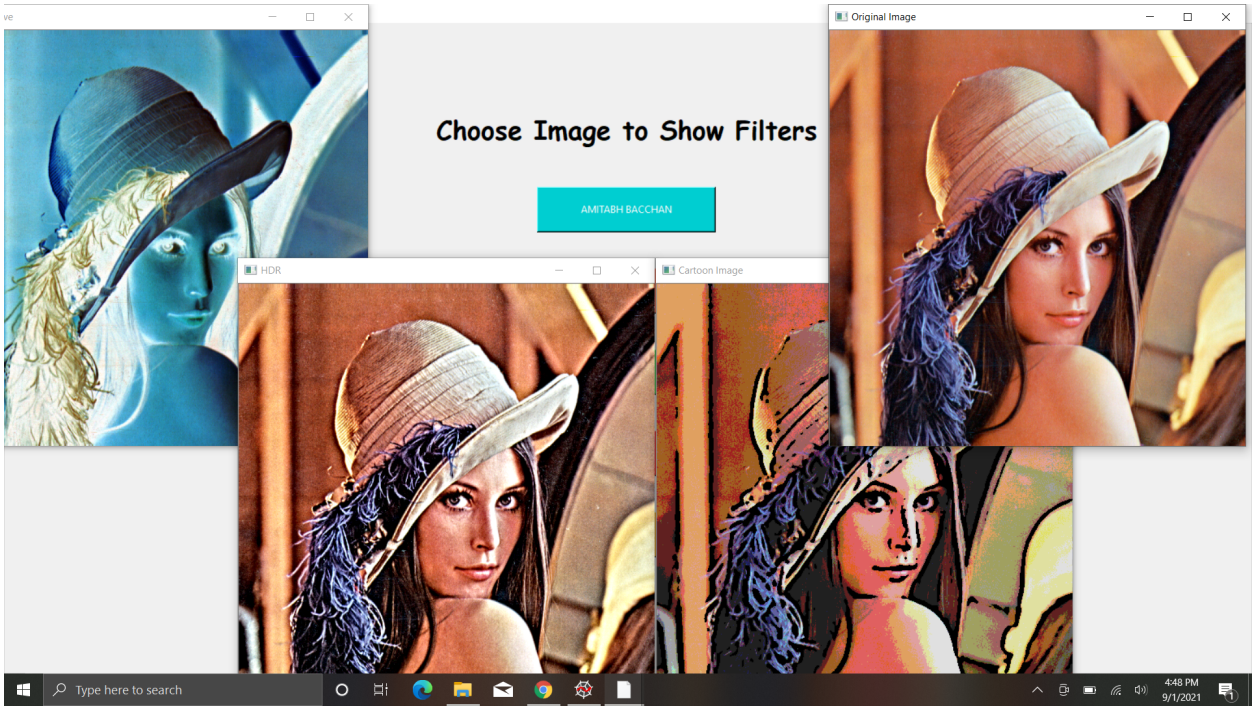
```
    # Convert the mask image back to color
    inverted_Bilateral   cv2.cvtColor(inverted_Bilateral, cv2.COLOR_GRAY2RGB)

    # Combine the edge image and the binned image
    cartoon_Bilateral   cv2.bitwise_and(inverted_Bilateral, img_bins)

    # Save the image
    cv2.imwrite('CartoonImage.png', cartoon_Bilateral)
    img33   cv2.imread('CartoonImage.png')

    negative   cv2.bitwise_not(img)
    hdr   cv2.detailEnhance(img, sigma_s 12, sigma_r 0.15)

    #show
    cv2.imshow('Negative', negative)
    cv2.imshow('HDR', hdr)
    cv2.imshow('Original Image', img)
    cv2.imshow('Cartoon Image', img33)


def   ()
    img   cv2.imread('pic.jpg')
    filter(img)


def         ()
    img   cv2.imread('meme_guy.jpg')
    filter(img)


def      ()
    img   cv2.imread('italy.jpg')
    filter(img)


def         ()
    img   cv2.imread('download.jpg')
    filter(img)


def           ()
    img   cv2.imread('hera_pheri.jpg')
    filter(img)

def      ()
    img   cv2.imread('lenna.jpg')
    filter(img)


ab   tkinter.Button(text 'AMITABH BACCHAN', command ab, bg 'darkturquoise',
fg 'white', width 30, height 3)
canvas1.create_window(500, 200, window ab)
```

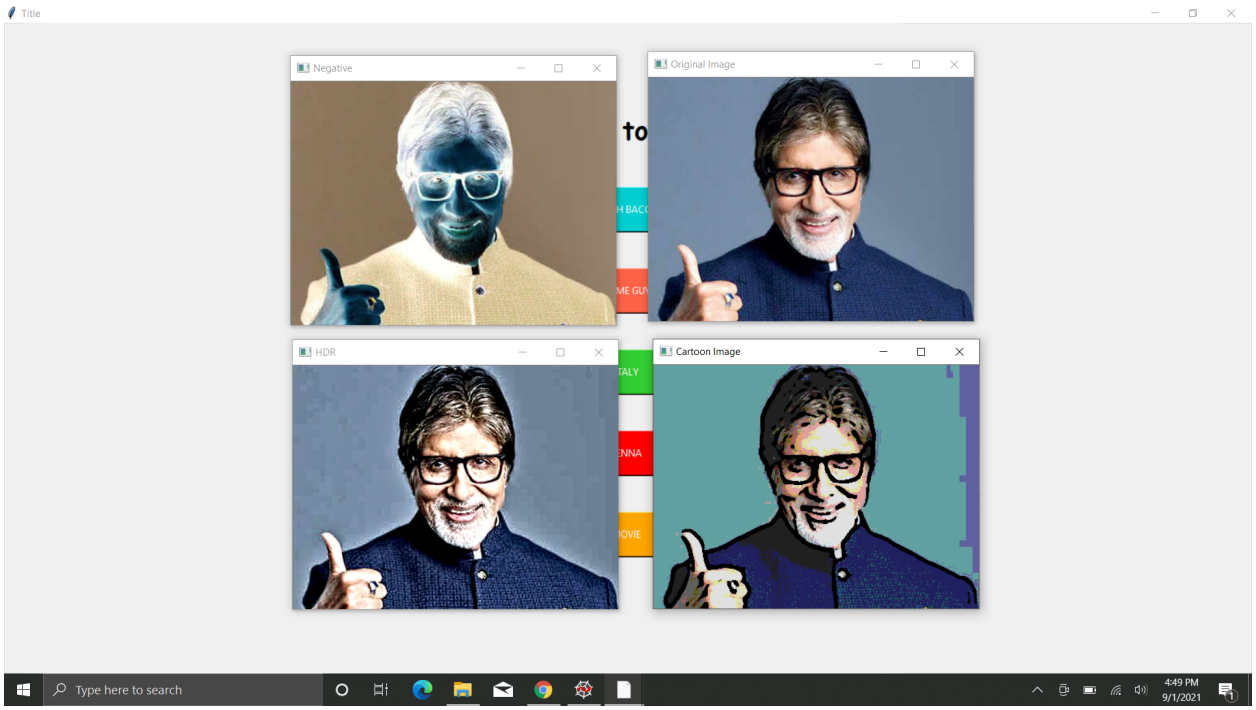
```
guy   tkinter.Button(text 'MEME GUY', command meme_guy, bg 'tomato',
fg 'white', width 30, height 3)
canvas1.create_window(500, 300, window guy)

it   tkinter.Button(text 'ITALY', command italy, bg 'lime green', fg 'white',
width 30, height 3)
canvas1.create_window(500, 400, window it)

le   tkinter.Button(text 'LENNA', command leena, bg 'red', fg 'white',
width 30, height 3)
canvas1.create_window(500, 500, window le)

hp   tkinter.Button(text 'MOVIE', command hera_pheri, bg 'orange', fg 'white',
width 30, height 3)
canvas1.create_window(500, 600, window hp)

root.mainloop()
```

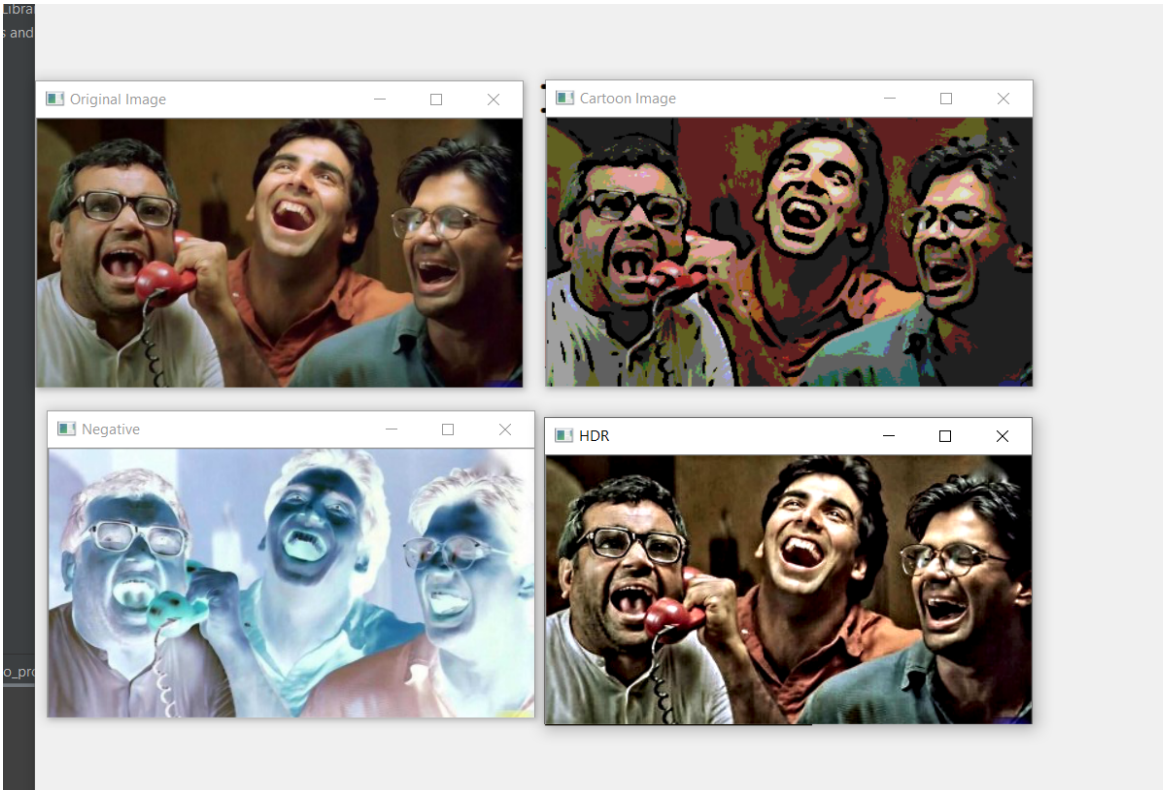
OUTPUT :




Cartoon Image
Original Image
Type here to search
4:47 PM
9/1/2021


Choose Image to Show Filters
AMITABH BACCHAN
HDR
Cartoon Image
Original Image
Type here to search
4:48 PM
9/1/2021


Title
Negative
Original Image
HDR
Cartoon Image
Type here to search
4:49 PM
9/1/2021


Original Image
Cartoon Image
Negative
HDR


Negative
Original Image
HDR
Cartoon Image
Type here to search
4:56 PM
9/1/2021