Name: Vikas Mane

Roll: B1921152

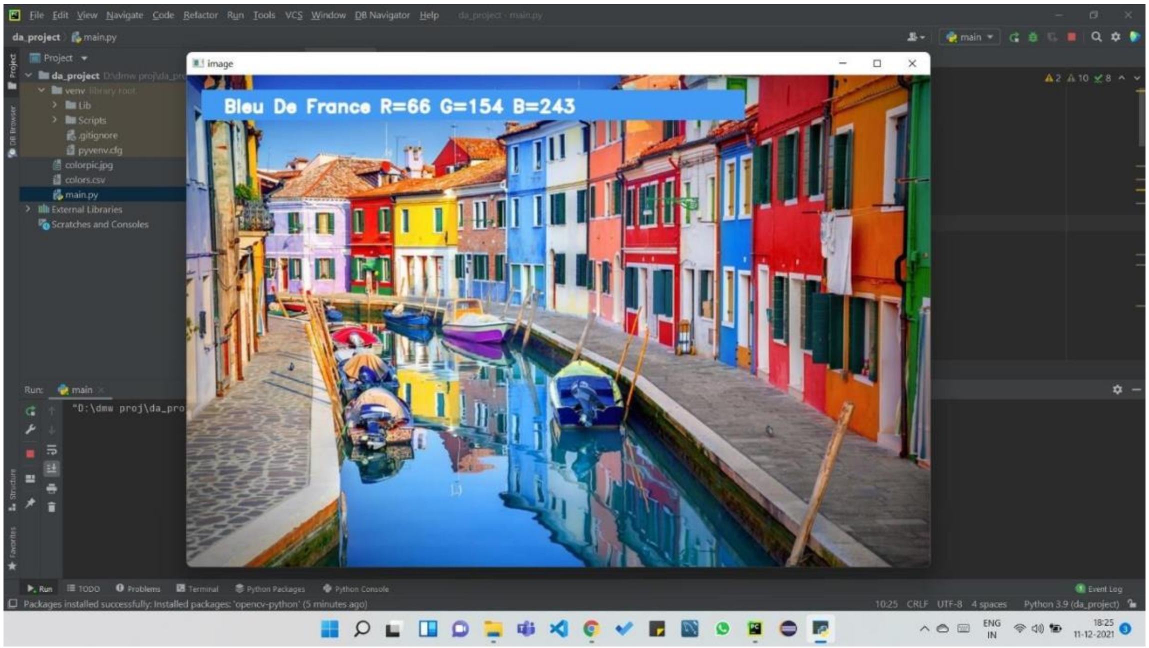
Color-detector.py

```
import cv2
import numpy as np
import pandas as pd
import argparse

# Creating argument parser to take image path from command line
ap = argparse.ArgumentParser()
ap.add_argument('-i', '--image', required=True, help="Image Path")
args = vars(ap.parse_args())
img_path = args['image']

# Reading the image with opencv
img = cv2.imread('D:\\dmw proj\\da_project\\colorpic.jpg')
# declaring global variables (are used later on)
clicked = False
r = g = b = xpos = ypos = 0

# Reading csv file with pandas and giving names to each column
index = ["color", "color_name", "hex", "R", "G", "B"]
csv = pd.read_csv('colors.csv', names=index, header=None)


# function to calculate minimum distance from all colors and get the most
matching color
def getColorName(R, G, B):
    minimum = 10000
    for i in range(len(csv)):
        d = abs(R - int(csv.loc[i, "R"])) + abs(G - int(csv.loc[i, "G"])) +
abs(B - int(csv.loc[i, "B"]))
        if (d <= minimum):
            minimum = d
            cname = csv.loc[i, "color_name"]
    return cname


# function to get x,y coordinates of mouse double click
def draw_function(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDBLCLK:
        global b, g, r, xpos, ypos, clicked
        clicked = True
        xpos = x
        ypos = y
        b, g, r = img[y, x]
        b = int(b)
        g = int(g)
        r = int(r)


cv2.namedWindow('image')
cv2.setMouseCallback('image', draw_function)
```

```
while (1):

    cv2.imshow("image", img)
    if (clicked):

        # cv2.rectangle(image, startpoint, endpoint, color, thickness)-1
fills entire rectangle
        cv2.rectangle(img, (20, 20), (750, 60), (b, g, r), -1)

        # Creating text string to display( Color name and RGB values )
        text = getColorName(r, g, b) + ' R=' + str(r) + ' G=' + str(g) + '
B=' + str(b)

        # cv2.putText(img,text,start,font(0-
7),fontScale,color,thickness,lineType )
        cv2.putText(img, text, (50, 50), 2, 0.8, (255, 255, 255), 2,
cv2.LINE_AA)

        # For very light colours we will display text in black colour
        if (r + g + b >= 600):
            cv2.putText(img, text, (50, 50), 2, 0.8, (0, 0, 0), 2,
cv2.LINE_AA)

        clicked = False

    # Break the loop when user hits 'esc' key
    if cv2.waitKey(20) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```

output:

1. cursor pointing at the blue wall



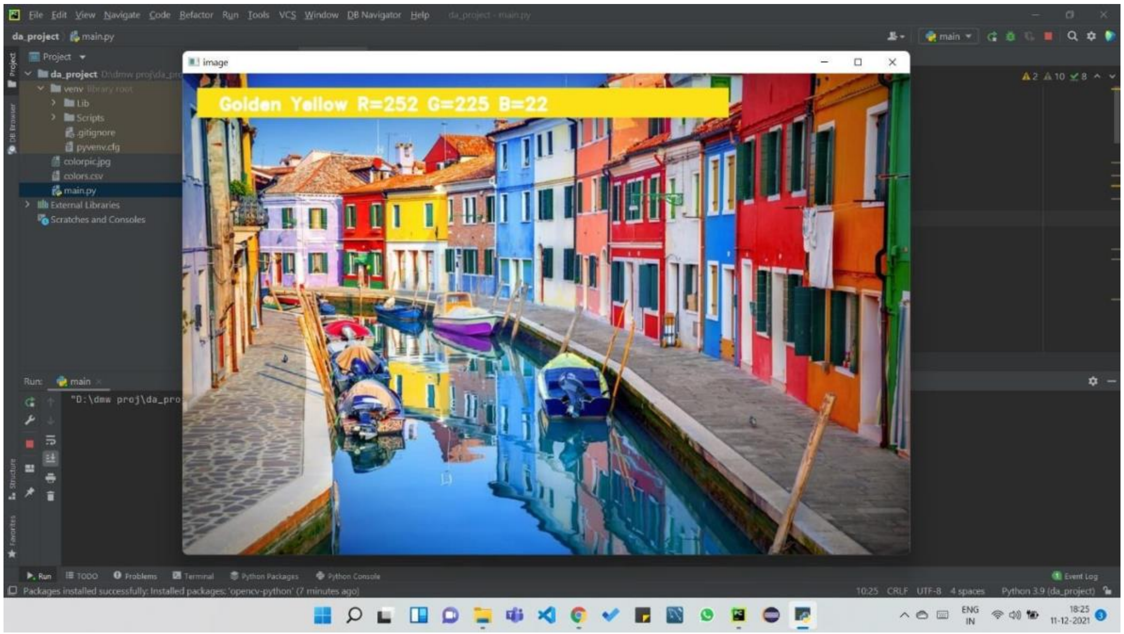
```
Bleu De France R=66 G=154 B=243
```

```
"D:\dmw proj\da_pro
```

```
Packages installed successfully: Installed packages: 'opencv-python' (5 minutes ago)
```

File Edit View Navigate Code Refactor Run Tools VCS Window DB Navigator Help da_project - main.py
da_project main.py
Project
da_project D:\dmw proj\da_pro
venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
colorpic.jpg
colors.csv
main.py
External Libraries
Scratches and Consoles
image
Run: main
Run TODO Problems Terminal Python Packages Python Console
Event Log
10:25 CRLF UTF-8 4 spaces Python 3.9 (da_project)
ENG IN
18:25 11-12-2021

2. Cursor pointing at the yellow wall



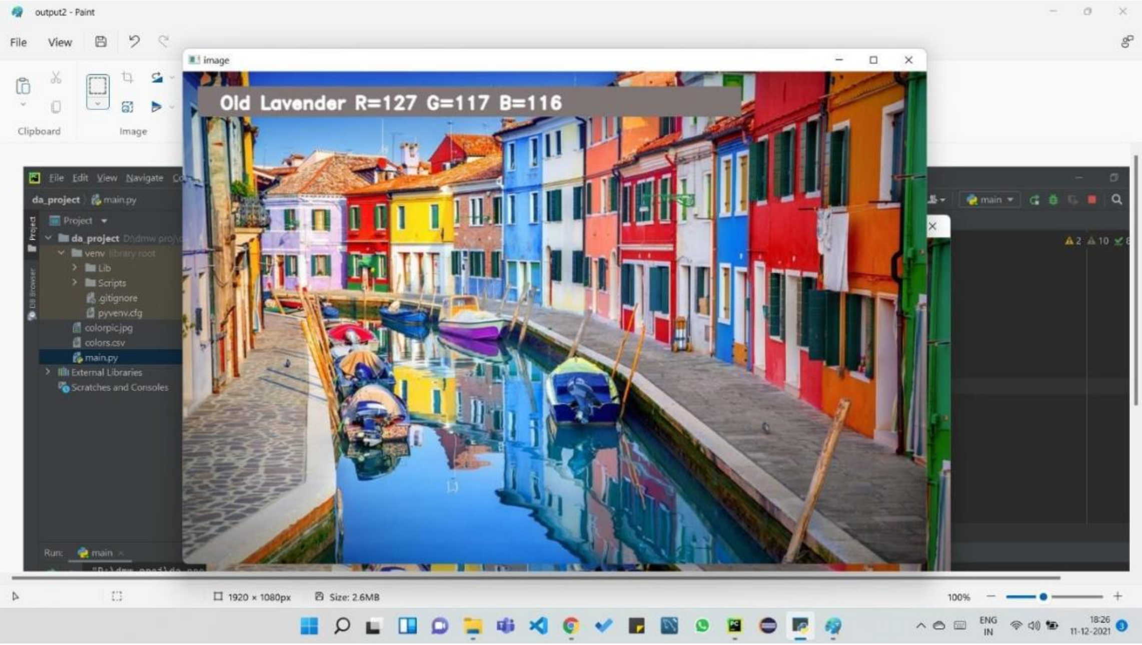
```
Golden Yellow R=252 G=225 B=22
```

```
"D:\dmw proj\da_pro
```

```
Packages installed successfully: Installed packages: 'opencv-python' (7 minutes ago)
```

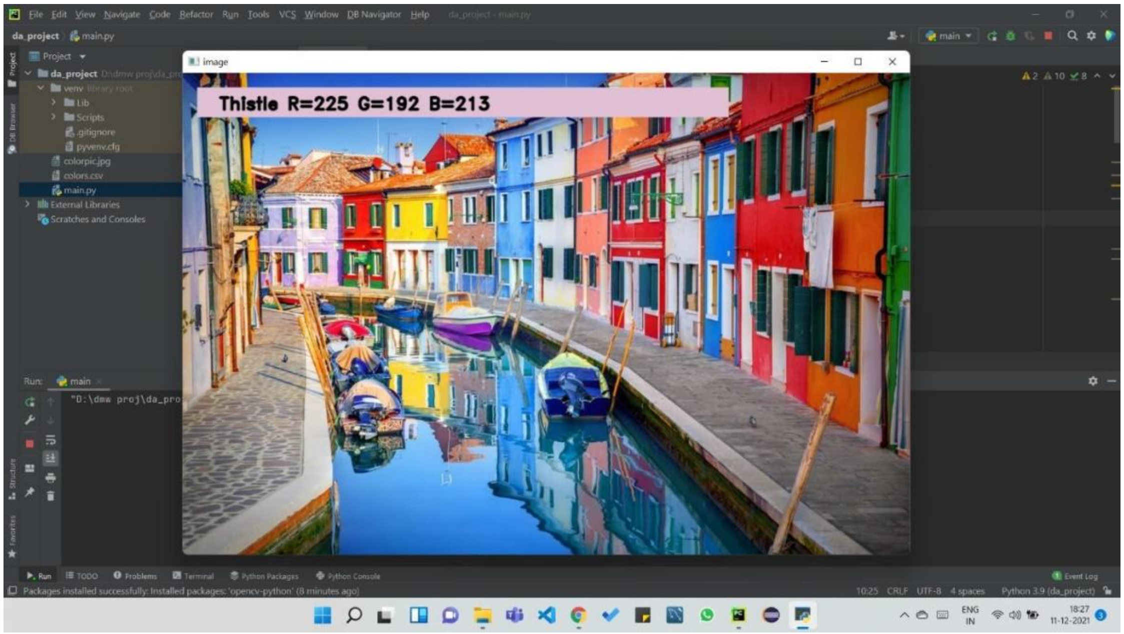
File Edit View Navigate Code Refactor Run Tools VCS Window DB Navigator Help da_project - main.py
da_project main.py
Project
da_project D:\dmw proj\da_pro
venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
colorpic.jpg
colors.csv
main.py
External Libraries
Scratches and Consoles
image
Run: main
Run TODO Problems Terminal Python Packages Python Console
Event Log
10:25 CRLF UTF-8 4 spaces Python 3.9 (da_project)
ENG IN
18:25 11-12-2021

3. cursor pointing at the grey colour



```
Old Lavender R=127 G=117 B=116
```

4. cursor pointing at the



```
Thistle R=225 G=192 B=213
```

```
"D:\dmw proj\da_pro
```

Packages installed successfully: Installed packages: 'opencv-python' (8 minutes ago)