

Recommendation system using computer vision



AXIOM CLOUD SOLUTION

Project by

- AMALA SEBESTIAN

PROBLEM STATEMENT

- While shopping, we totally neglect the fact that **not every type of brown skin can compliment colors shown on fair-skins**. Models wearing varieties of ethnic clothes and showing how lovely they look in them make us add every such item in the shopping cart. But little do we know that these models on Indian e-commerce websites are foreigners – on whom every vibrant color of Indian ethnicity looks pretty awesome.
- Browsing through various shopping portals in hope of finding the best coloured dress that suits the skin tone according to the occasion is a **time consuming activity**.

The Idea

- Here we are introducing our solution to this problem . Our mission is to recognize indian skin tones in the global market and get best dresses for them according to their skin tone .

Following are the field of interest which we are providing to users

- AGE
- GENDER
- OCCASSION
- SKIN TONE

HOW WE ARE MAKING THINGS EASY

- Currently we are using a **self made database**, which is based on **combination ,occassion,vote_avg, vote_count ,gender ,skin colour ,prices and o_p [links where the clothes will be available]**

J1	A	B	C	D	E	F	G	H	I	J	K
1	index	combination	occasion	vote_avg	vote_count	gender	skin_colour	combination	prices	o_p	
2	1	WHITE SHIRT WITH BLACK TRO	INTERVIEW	6.5	7577	MALE	MEDIUM FAIR	WHITE SHIRT WITH	300	www.flipkart.com	
3	2	LIGHT BLUE SHIRT WITH BLACK	INTERVIEW	8.9	8638	MALE	MEDIUM FAIR	LIGHT BLUE SHIRT \	350	www.amazon.com	
4	3	WHITE SHIRT WITH BLACK TRO	INTERVIEW	9	468	MALE	MEDIUM FAIR	WHITE SHIRT WITH	400	www.flipkart.com	
5	4	WHITE SHIRT WITH BROWN TR	INTERVIEW	6.8	9386	MALE	MEDIUM FAIR	WHITE SHIRT WITH	500	www.flipkart.com	
6	5	LIGHT BLUE SHIRT WITH DARK	INTERVIEW	8.6	753	MALE	MEDIUM FAIR	LIGHT BLUE SHIRT \	350	www.amazon.com	
7	6	SKY BLUE SHIRT WITH BROWN	INTERVIEW	9.8	8368	MALE	MEDIUM FAIR	SKY BLUE SHIRT WI	670	www.amazon.com	
8	7	PINK SHIRT WITH BLACK TROU	INTERVIEW	7.1	3863	MALE	MEDIUM FAIR	PINK SHIRT WITH B	789	www.flipkart.com	
9	8	LIGHT BLUE SHIRT WITH BLACK	INTERVIEW	7.3	5848	MALE	MEDIUM FAIR	LIGHT BLUE SHIRT \	999	www.amazon.com	
10	9	GREY SHIRT WITH BROWN TRO	INTERVIEW	6.2	3763	MALE	MEDIUM FAIR	GREY SHIRT WITH	150	www.flipkart.com	

We will be using a big database from differnt websites which are based on price, ratings, blogs,trend and many more we are going to add.

in future

HOW WE ARE MAKING THINGS EASY

we have used python language and its libraries namely:

pandas pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

numpy NumPy is the fundamental package for scientific computing in Python. ... NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data

we are using streamlit for creating the webapp

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science

we have also used computer vision opencv to live detect the skin colour

cv2 OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

Codes

- in order to get the top rated combination for all skin type we used a formula:

$$(r \cdot v + c \cdot m) / (v + m)$$

- where $r = df_m['vote_avg']$
- $v = df_m['vote_count']$
- $c = df_m['vote_avg'].mean()$

```
1  from numpy.lib.twodim_base import mask_indices
2  import pandas as pd
3  from pandas.core.frame import DataFrame
4  import streamlit as st
5  import plotly.express as px
6  from PIL import Image
7  from streamlit.state.session_state import Value
8
9  st.set_page_config(page_title='TREND WITH AI')
10 st.header(' TREND with AI')
11 excel_file='sample.xlsx'
12
13 df_m=pd.read_csv(r'C:\Users\amala\Downloads\maindb.csv')
14 r=df_m['vote_avg']
15 v=df_m['vote_count']
16 c=df_m['vote_avg'].mean()
17 m=500
18 d=df_m['weighted_votes']=(r*v+c*m)/(v+m)
19 d=df_m[['occasion','combination','weighted_votes','skin_colour','prices','o_p']].sort_values('weighted_votes',ascending=False)
20 v=df_m[['occasion','combination','weighted votes','skin colour','prices','o p']]
```

- Now we made user selected skin colour's unique list and create a mask for it using numpy ,we also made an occassion based unique list and created a mask for it and at last we sorted the top rated combination for a particular skin_colour an displayed it on the web app

```
-->
26 skin=d['skin_colour'].unique().tolist()
27 skin_selected=['MEDIUM FAIR']
28 mask=d['skin_colour'].isin(skin_selected)
29 f=d[mask].head(60)
30 occ=f['occasion'].unique().tolist()
31 occ_selected=['INTERVIEW']
32 mask=f['occasion'].isin(occ_selected)
33 data1=f[mask].head(30)
34
35
36 skin=d['skin_colour'].unique().tolist()
37 skin_selected=['MEDIUM FAIR']
38 mask=d['skin_colour'].isin(skin_selected)
39 e=d[mask].head(60)
40 var=e['occasion'].unique().tolist()
41 var_selected=['FESTIVE']
42 mask=e['occasion'].isin(var_selected)
43 data2=e[mask].head(20)
44
45
46 skin=d['skin_colour'].unique().tolist()
47 skin_selected=['MEDIUM FAIR']
48 mask=d['skin_colour'].isin(skin_selected)
49 s=d[mask].head(60)
50 gar=s['occasion'].unique().tolist()
51 gar_selected=['FRESHERS']
```

- To add price filter we have used the following code :

The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar contains icons for Explorer, Search, Find, Split, and Settings. The Explorer view shows a workspace with several files and folders:

- python_course
- first.py
- main.py
- Downloads
- .streamlit
- config.toml
- AI.mp4
- Anaconda3-2021.05...
- ansa ppt (1).pdf
- ansa ppt (2).pdf
- ansa ppt.pdf
- app.py
- backend.xlsx
- ChromeSetup.exe
- Color detection.zip
- Cultural Club details...
- Cultural Club details...
- cv.py
- desktop.ini
- download.jpg
- Drk medium.xlsb.xlsx
- fes.pdf
- fest.jpg
- fresh.jpg
- fresh.pdf

The main editor tab is titled "app.py - Untitled (Workspace) - Visual Studio Code". The code in the editor is as follows:

```
52     mask=s['occasion'].isin(gar_selected)
53     data3=s[mask].head(20)
54
55
56
57     skin=v['skin_colour'].unique().tolist()
58     skin_selected=['MEDIUM FAIR']
59     mask=v['skin_colour'].isin(skin_selected)
60     g=v[mask].head(30)
61     lal=g['occasion'].unique().tolist()
62     lal_selected=['FRESHERS']
63     mask=g['occasion'].isin(lal_selected)
64     data4=g[mask].tail(10)
65     dul=data4.sort_values('prices',ascending= True)
66     hul=data4.sort_values('prices',ascending=False)
67
68
69     skin=k['skin_colour'].unique().tolist()
70     skin_selected=['MEDIUM FAIR']
71     mask=k['skin_colour'].isin(skin_selected)
72     n=k[mask].head(60)
73     nal=n['occasion'].unique().tolist()
74     nal_selected=['INTERVIEW']
75     mask=n['occasion'].isin(nal_selected)
76     data7=n[mask].tail(20)
77     lul=data7.sort_values('prices',ascending= True)
78     nul=data7.sort_values('prices',ascending=False)
79
80
81
```

- WE CONNECTED THE REQUIRED TOP RATED DATABASE TO OUR WEBAPP USING THE FOLLOWING CODE:

app.py - Untitled (Workspace) - Visual Studio Code

```

Run Terminal Help app.py - Untitled (Workspace) - Visual Studio Code
ip.py X cv.py n config.toml
nloads > app.py



```

Terminal Help app.py - Untitled (Workspace) - Visual Studio Code

```

> app.py
nav=st.sidebar.radio("Navigation",["Home","Profile","Trends","About Us"])

if nav=="Home":
    nickname=st.text_input("WHAT'S YOUR NICKNAME?")

    st.write("welcome {}".format(nickname))

    st.write("HOME")
    image =Image.open(r'C:\Users\amala\Downloads\TREND with AI.jpg')
    st.image(image,caption='Let your style speaks for you',use_column_width=True)

    st.sidebar.radio("Gender",["Male","Female"])
    st.subheader("Want to know the TREND?")
    st.write("Select the checkbox, and let your style speak for you ")
    if st.checkbox("Show Table"):
        st.table(d)

skin=d['skin_colour'].unique().tolist()
skin_selection=st.multiselect('skin_colour:',skin,default=skin)
mask=(d['skin_colour'].isin(skin_selection))

st.subheader("Get customised dresses according to your occasion !!")

```

Terminal Help app.py - Untitled (Workspace) - Visual Studio Code

```

> app.py
choice =st.selectbox("Choose your occasion","","Freshers","Festive","Interview")

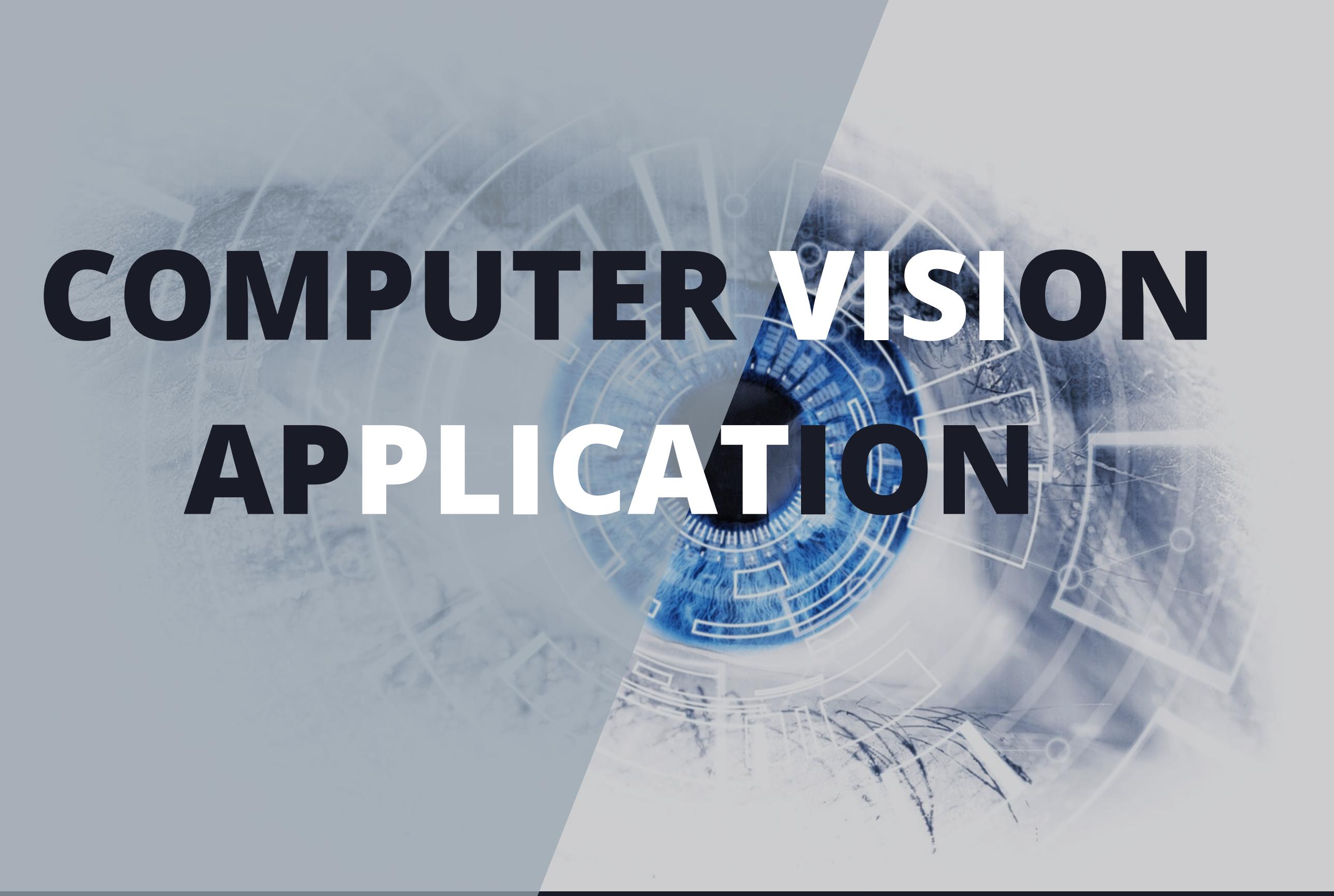
if choice == "Interview":
    st.image(img,caption='Recommendations for you ',use_column_width=True)
    st.table(data1)

    st.write("Who told you that Trendy dresses are costly ?")
    st.subheader("Now you can get YOUR PERFECT FIT at YOUR COST")
    temp_options =['Low','Medium','High']
    temp=st.select_slider("Choose your range",options=temp_options)
    if temp =='Low':
        st.dataframe(lul)
    if temp == 'High':
        st.dataframe(nul)

if choice == "Festive":
    st.image(fes,caption='Recommendations for you',use_column_width=True)
    st.table(data2)

```

COMPUTER VISION APPLICATION



- It opens the user's camera & and once the user takes a screenshot it returns their skin colour which can be used by the user to know the category under which their skin colour falls

app.py - Untitled (Workspace) - Visual Studio

```

File Go Run Terminal Help
.. app.py cam.py pro.py cv.py n
Downloads > app.py > ...
05...
189
190     import cv2
191     import numpy as np
192
193     cam=cv2.VideoCapture(0)
194     cv2.namedWindow(" Instant shots/ TrendwithAI")
195
196     img_counter=0
197
198
199     while True:
200         ret,frame=cam.read()
201
202         if not ret:
203             print("failed to grab frame")
204             break
205
206         cv2.imshow("test",frame)
207
208
209         k=cv2.waitKey(1)
210
211         if k%256 ==27:
212             print("Escape hit,closing the app")
213             break
214
215         elif k%256==32:
216             img_name="fash{}.jpg".format(img_counter)
217             cv2.imwrite(img_name,frame)
218             print("Screenshot taken")

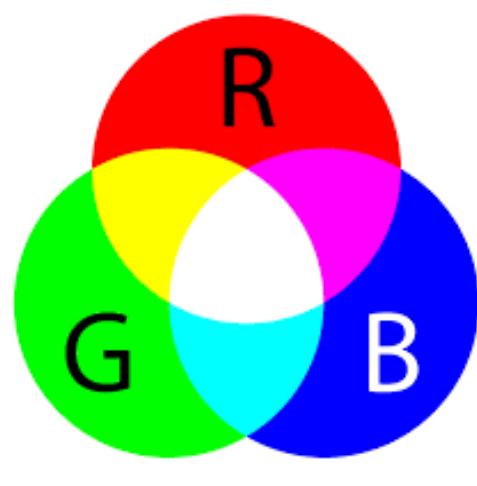
```

app.py x cam.py pro.py cv.py n config.toml

```

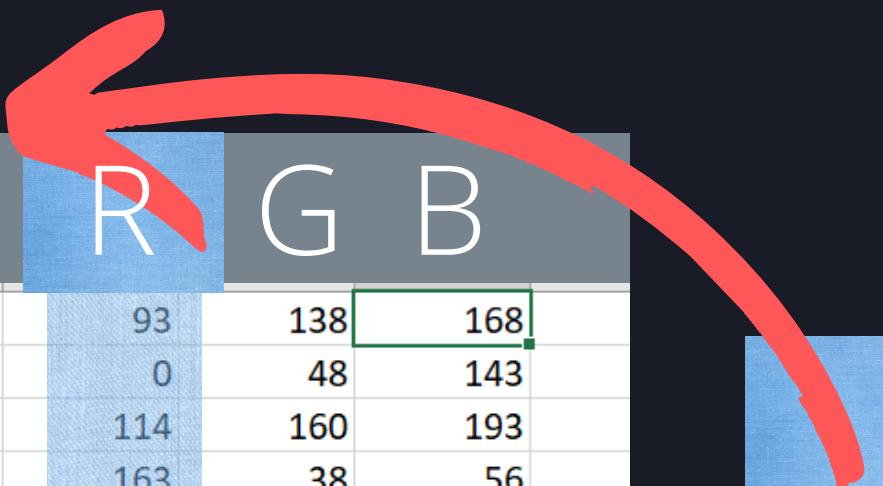
Downloads > app.py > ...
8
9     print("Screenshot taken")
10    img_counter+=1
11
12    img_path = r'fash1.jpg'
13    img = cv2.imread(img_path)
14
15    # declaring global variables (are used later on)
16    clicked = False
17    r = g = b = x_pos = y_pos = 0
18
19    # Reading csv file with pandas and giving names to each column
20    index = ["color", "color_name", "hex", "R", "G", "B"]
21    csv = pd.read_csv(r'C:\Users\amala\Downloads\colors.csv', names=index, header=None)
22
23
24    # function to calculate minimum distance from all colors and get the most matching color
25    def get_color_name(R, G, B):
26        minimum = 10000
27        for i in range(len(csv)):
28            d = abs(R - int(csv.loc[i, "R"])) + abs(G - int(csv.loc[i, "G"])) + abs(B - int(csv.loc[i, "B"]))
29            if d <= minimum:
30                minimum = d
31                cname = csv.loc[i, "color_name"]
32
33        return cname
34
35
36    # # function to get x,y coordinates of mouse double click
37    def draw_function(event, x, y, flags, param):
38        if event == cv2.EVENT_LBUTTONDOWN:

```



colour

			R	G	B
1	air_force_blue_raf	Air Force Blue (Raf)	#5d8aa8	93	138
2	air_force_blue_usaf	Air Force Blue (Usaf)	#00308f	0	48
3	air_superiority_blue	Air Superiority Blue	#72a0c1	114	160
4	alabama_crimson	Alabama Crimson	#a32638	163	38
5	alice_blue	Alice Blue	#f0ff8f	240	248
6	alizarin_crimson	Alizarin Crimson	#e32636	227	38
7	alloy_orange	Alloy Orange	#c46210	196	98
8	almond	Almond	#efdec0	239	222
9	amaranth	Amaranth	#e52b50	229	43
10	amber	Amber	#ffbf00	255	191
11	amber_sae_ece	Amber (Sae/Ece)	#ff7e00	255	126
12	american_rose	American Rose	#ff033e	255	3
13	amethyst	Amethyst	#96c	153	102
14	android_green	Android Green	#a4c639	164	198
15	anti_flash_white	Anti-Flash White	#f2f3f4	242	243
16	antique_brass	Antique Brass	#cd9575	205	149
17	antique_fuchsia	Antique Fuchsia	#915c83	145	92
18	antique_ruby	Antique Ruby	#841b2d	132	27
19	antique_white	Antique White	#faebd7	250	235
20	ao_english	Ao (English)	#008000	0	128



$\text{Abs}(R - \text{int}(\text{csv.loc}[l, "R"])) + \text{Abs}(G - \text{int}(\text{csv.loc}[l, "G"])) + \text{Abs}(B - \text{int}(\text{csv.loc}[l, "B"]))$

```
op.py cam.py pro.py cv.py config.toml

Downloads > app.py > ...
# # function to get x,y coordinates of mouse double click
def draw_function(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        global b, g, r, x_pos, y_pos, clicked
        clicked = True
        x_pos = x
        y_pos = 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 True:
    cv2.imshow("image", img)
    if clicked:
        # cv2.rectangle(image, start point, 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 = get_color_name(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)
        (variable) b: Literal[0]

    cv2.putText(img, text, (50, 50), 2, 0.8, (255, 255, 255), 2, cv2.LINE_AA)

    cv2.waitKey(1)
```

```
Terminal Help • app.py - Untitled (Workspace) - Visual Studio Code
/ ● cam.py pro.py cv.py ⌂ n config.toml
ds > app.py > ...
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

cam.release()
# cam.destroyAllWindows()
cv2.destroyAllWindows()

st.write("Find your sjin colour in the given table & FLAUNT IT !")

tone=pd.read_csv(r'C:\Users\amala\Downloads\tone.csv')

if st.checkbox("Show Table"):
    st.table(tone)

if nav=="About Us":
    st.subheader("About us")
    st.write("People are not made for dresses, but Dresses are made for the people. With that
```

CONCLUSION

Every thing has its own pros and cons

BENEFITS:-

- Time saving
- Best recommendation
- Better enviroment
- Covnvenient way

THINGS TO BE WORK ON:-

- Larger data base
- More filters during selection
- As the lighting can affect the correct detection of skin colour
- A/B testing will be required to do,T test and so on

Future Scope

- The skin tone detection feature of our webapp can be used in cosmetic E-commerce websites.
- More work can be done in this feature to detect pores and acne and get customised solutions for it
- More skin tones category can be made available to the user to get personalised clothing.