

```

1  ''' Weather Application By
2
3      R/BCA-20-225
4      R/BCA-20-104
5      R/BCA-20-221
6
7  Features: Gets weather data scraped from timeanddate.com of any state of India.
8  Displays temperature
9  Displays Type of Weather
10 Displays a picture depending on the type of weather
11 Background colour changes according to type of weather
12 used tkinter, PIL and BeautifulSoup library
13 '''
14
15
16
17
18
19 from tkinter import *
20 from PIL import ImageTk,Image
21 from bs4 import BeautifulSoup
22 import requests
23
24 root=Tk()
25 root.geometry('400x400')
26 root.title('Weather App')
27 root.resizable(width=False, height=False)
28
29 def hide_frames():
30     for widget in search_frame.winfo_children():
31         widget.destroy()
32     search_frame.pack_forget()
33     label_img.pack()
34     e.delete(0,'end')
35
36 def search():
37     try:
38         global search_frame
39         label_img.pack_forget()
40
41         search_frame=LabelFrame(root,bg='black',width=400,height=400)
42         search_frame.pack(ipadx=200,ipady=200)
43
44
45         #scrape
46         place=e.get()
47         if place=='':
48             place_text='India'
49         else:
50             place_text=place.title()
51
52
53
54
55         html_text=requests.get('https://www.timeanddate.com/weather/india/'+place).text
56         soup = BeautifulSoup(html_text, 'lxml')
57         place=soup.find('h1',class_='headline-banner__title')
58         container=soup.find('div',class_='bk-focus__qlook')
59         w_type=container.find_all('p')
60         temp=container.find('div',class_='h2')
61         container2=soup.find('table',class_='table table--left table--inner-borders-rows')
62
63         weather_type_text=w_type[1].text
64         print(weather_type_text)

```

```

65         temp_text=temp.text
66         #scrape-end
67
68         #labels
69         place_label=Label(search_frame,text=place_text,font=
('helvetica',13,'bold'),anchor="center")
70         place_label.pack(pady=20)
71         global label_img1
72         bakgrnd='white'
73         w_type_descision='partlycloudy.png'
74         if 'Clear' in weather_type_text:
75             w_type_descision='partlysunny.jpg'
76             bakgrnd='yellow3'
77             print('1')
78
79         elif 'Partly cloudy' in weather_type_text or 'Passing clouds'in weather_type_text:
80             w_type_descision='partlycloudy.png'
81             bakgrnd='blue3'
82             print('2')
83
84         elif 'Sunny'in weather_type_text:
85             w_type_descision='sunny.jpg'
86             bakgrnd='yellow'
87             print('3')
88
89         elif 'Haze' in weather_type_text or 'Fog' in weather_type_text or 'Clouds' in
weather_type_text:
90             w_type_descision='cloudy.jpg'
91             bakgrnd='blue4'
92             print('4')
93
94         search_frame.config(bg=bakgrnd)
95
96
97         label_img1=ImageTk.PhotoImage(Image.open(w_type_descision))
98         label_img1=Label(search_frame,image=label_img1)
99         label_img1.pack(pady=20)
100
101
102         weather_type_label=Label(search_frame,anchor="center",text=weather_type_text,font=
('helvetica',15),bg='white')
103         weather_type_label.pack(ipadx=200)
104
105         temp_label=Label(search_frame,text=temp_text,anchor="center",font=
('helvetica',15),bg='white',bd=1)
106         temp_label.pack(ipadx=50)
107         #labels
108         button2=Button(search_frame,text='back',command=hide_frames).pack(pady=20)
109     except:
110         print("Error ocured,re-open app")
111
112     #labels2
113     label_img=ImageTk.PhotoImage(Image.open('weather_app.jpg'))
114     global label_img
115     label_img=Label(image=label_img)
116     label_img.pack()
117
118     #Entry and buttons
119     e = Entry(root,font=('helvetica',14),bd=2)
120     e.place(x = 72, y = 242)
121     button=Button(root,text='Get Details',command=search).place(x = 165, y = 300)
122
123     root.mainloop()
124

```

