# PROJECT CODE

```
from tkinter import *
from tkinter.ttk import *
from tkinter.messagebox import showerror

import plotly.graph_objects as go

# Load data frame and tidy it.
import pandas as pd
path="C:\\wander project\\"

window = Tk()

#global variables
name=""
month=""
state=""
info=""
style = Style()


#function to clear screen
def clearScreen():
    list = window.pack_slaves()
    for l in list:
l.destroy()

#functions called on button clicks
def detailClicked():
    global name
    global month

    name=txt1.get()
```

```
    month=txt2.get()
    month=month.lower()
    if month=="":
showerror(title='input  missing',message='Please  enter
month')
    else :
clearScreen()
        printScreen3()

def beginClicked():
clearScreen()
    printScreen2()

def nextClicked():
clearScreen()
    printScreen4()

def stateClicked():
    global state

    state=txt3.get()
    state=state.capitalize()

    if state=="" :
showerror(title='input  missing',message='Please  enter
state')
    else:
clearScreen()
        printScreen5()

def lastClicked():
clearScreen()
    printScreen6()

#functions for printing screens
def printScreen1():
```

```
style.configure("BW.TLabel",        foreground="white",
background="slate grey",
                              font= 'Helvatica 24 bold',
borderwidth=20,width= 24, anchor=CENTER,relief=RAISED)
    label = Label(top_frame, image = img1).pack()
btn = Button(bottom_frame, text = "Let the adventure
begin..",style="BW.TLabel",command=beginClicked).pack(
side = "bottom")


def printScreen2():
    label = Label(top_frame, image = img2).pack()
    # style configuration for label
style.configure("Label", font= 'Helvatica 10 bold')

    lbl1 = Label(bottom_frame, text="Please enter your
name ",style="Label").pack()
    txt1.pack()
    lbl2 = Label(bottom_frame, text="In which month do
you want to travel ",style="Label").pack()
    txt2.pack()

    # style configuration for button
style.configure("BW.TLabel",        foreground="white",
background="slate grey",
                             font=  'Helvatica  12  bold',
borderwidth=5,width= 10, anchor=CENTER,relief=RAISED)
btn                =                Button(bottom_frame,
text="Next..",style="BW.TLabel",
command=detailClicked).pack(side="bottom")

def printScreen3():
    label = Label(top_frame, image = img3).pack()
    # style configuration for button
style.configure("BW.TLabel",        foreground="white",
background="cyan4",
```

```
                            font= 'Helvatica 12 bold',
borderwidth=5,width= 40, anchor=CENTER,relief=RAISED)
      btn1 = Button(bottom_frame, text = "Temperature
pattern..",style="BW.TLabel",command=plotTempMap).pack
(side = "top")

     btn2 = Button(bottom_frame, text = "Precipitation
pattern..",style="BW.TLabel",command=plotPptMap).pack(
side = "top")

       btn3 = Button(bottom_frame, text = "Landform
Distribution..",style="BW.TLabel",command=plotLandMap)
.pack(side = "top")


             btn5   =   Button(bottom_frame,   text   =
"Next..",style="BW.TLabel",command=nextClicked).pack(s
ide = "bottom")


def printScreen4():
    label = Label(top_frame, image = img4).pack()
    global name

    # style configuration for label
style.configure("Label", font= 'Helvatica 10 bold')
    text1= "Dear " +name + ",Enter name of state "
                    lbl1     =     Label(bottom_frame,
text=text1,style="Label").pack()

    txt3.pack()
    # style configuration for button
style.configure("BW.TLabel",        foreground="white",
background="slate grey",
                            font= 'Helvatica 12 bold',
borderwidth=5,width= 10, anchor=CENTER,relief=RAISED)
```

```
btn                =                Button(bottom_frame,
text="Next..",style="BW.TLabel",
command=stateClicked).pack(side=BOTTOM)

def printScreen5():
    label = Label(top_frame, image = img5).pack()
    global state
    global info

     # style configuration for label
style.configure("Label",  font=  'Helvatica  10  bold',
space=4)
consolinfo()
       lbl1  =  Label(bottom_frame,  text="Consolidated
Information          about          "+state+"\n"+info,
style="Label").pack()

    # style configuration for button
style.configure("BW.TLabel",        foreground="white",
background="slate grey",
                           font=  'Helvatica  12  bold',
borderwidth=5,width= 10, anchor=CENTER,relief=RAISED)
btn                =                Button(bottom_frame,
text="Next..",style="BW.TLabel",command=lastClicked).p
ack(side=BOTTOM)

def printScreen6():
    label = Label(window, image = img6).pack()

#functions for plotting maps
def plotTempMap():
        global month

df = pd.read_csv(path+'US_monthly_temperature.csv')

        fig = go.Figure(data=go.Choropleth(
```

```
                    locations=df['code'],  #  Spatial
coordinates
            z = df[month], # Data to be color-coded
locationmode = 'USA-states', # set of locations match
entries in `locations`
colorscale = 'thermal',
colorbar_title = "Temperature(°F)",
hoverinfo= "all",
            text=df['state']
        ))

fig.update_layout(
title_text = 'US temperature by State',
geo_scope='usa', # limite map scope to USA
        )

fig.show()

def plotPptMap():
df = pd.read_csv(path+'US_monthly_precipitation.csv')

        fig = go.Figure(data=go.Choropleth(
                    locations=df['code'],  #  Spatial
coordinates
             z = df[month].astype(float), # Data to be
color-coded
locationmode = 'USA-states', # set of locations match
entries in `locations`
colorscale = 'teal',
colorbar_title = "Precipitation",
hoverinfo= "all",
            text=df['state']
        ))

fig.update_layout(
title_text = 'US precipitation by State',
```

```
geo_scope='usa', # limite map scope to USA
        )

fig.show()

def plotLandMap():
df = pd.read_csv(path+'US_landforms.csv')

        fig = go.Figure(data=go.Choropleth(
                        locations=df['code'],  #  Spatial
coordinates
            z = df['tc'], # Data to be color-coded
locationmode = 'USA-states', # set of locations match
entries in `locations`
colorscale = 'sunset',
colorbar_title = "Landforms",
hoverinfo= "all",
            text=df['Landform'],
showscale=False
        ))

fig.update_layout(
title_text = 'US landforms by State',
geo_scope='usa', # limite map scope to USA
        )

fig.show()


#function for fetching consolidated info for a state
def consolinfo():
        global info
df = pd.read_csv(path+'consolidated.csv')
         info="Landform type :" + df[state].values[0] +
"\n"    +    "Annual    high    temperature    :"    +
df[state].values[1]  +  "\n"+"Annual  low  temperature
```

```
:"+df[state].values[2]  +  "\n"+"Annual  precipitation
:"+df[state].values[3]  +  "\n"+"Best  place  to  visit
:"+df[state].values[4] + "\n"


window.title("Welcome to WanderUSA")
window.geometry('620x640')
window.resizable(0, 0)

top_frame = Frame(window).pack()
bottom_frame = Frame(window).pack(side = "bottom")

# show image in top frame
img1    =    PhotoImage(file    =    path+"\\pictures
project\\wonder wander repeat.png")
img2    =    PhotoImage(file    =    path+"\\pictures
project\\img2.png")
img3    =    PhotoImage(file    =    path+"\\pictures
project\\img3.png")
img4    =    PhotoImage(file    =    path+"\\pictures
project\\img4.png")
img5    =    PhotoImage(file    =    path+"\\pictures
project\\img5.png")
img6    =    PhotoImage(file    =    path+"\\pictures
project\\img6.png")

txt1 = Entry(window,width=20,justify=CENTER)
txt2 = Entry(window,width=20,justify=CENTER)
txt3 = Entry(window,width=20,justify=CENTER)
printScreen1()

window.mainloop()
```