In [11]:

```
import speech_recognition as sr
import pyttsx3
from tkinter import *
import tkinter.messagebox
import tkinter.font as tkfont
from PIL import ImageTk, Image
import pickle
import import_ipynb
import numpy as np

base = Tk()
base.title("Hello")
base.geometry("400x500")
base.resizable(width=FALSE, height=FALSE)

image = ImageTk.PhotoImage(Image.open("pic9.png").resize((500, 500)))
#canvas = Canvas(base, width=400, height=500)
#canvas.pack()
#canvas.background = image
#canvas.create_image(0, 0, anchor=NW, image=image)

import pandas as pd
yield_df = pd.read_csv("yield_df.csv")
yield_df.drop(columns = ['Unnamed: 0','Year'], inplace=True)
country_list = list(yield_df.Area)
item_list = list(yield_df.Item)

crop_df = pd.read_csv('cpdata.csv')
crop_df.drop(columns = ['Unnamed: 0'], inplace=True)

from sklearn.preprocessing import MinMaxScaler, StandardScaler
scaler1 = MinMaxScaler()
scaler2 = StandardScaler()

model_yield = pickle.load(open('yield_model.sav','rb'))
model_crop = pickle.load(open('crop_model.sav', 'rb'))

crops = ['Adzuki Beans', 'Apples', 'Bananas', 'Black gram', 'Chick peas',
       'Coconut', 'Coffee', 'Cotton', 'Groundnuts', 'Jute', 'Kidney Beans',
       'Lentils', 'Mangoes', 'Moth Beans', 'Mung Bean', 'Oranges', 'Papayas',
       'Peas', 'Pigeon peas', 'Rubber', 'Sugar cane', 'Tea', 'Tobacco',
       'Watermelons', 'grapes', 'maize', 'millet', 'muskmelon', 'pomegranate',
       'rice', 'wheat']

f1 = Label(base, image=image)
f2 = Label(base, image=image)
f3 = Label(base, image=image)

#f1 = Frame(base, width="400", height="500")
#f2 = Frame(base, width="400", height="500")
#f3 = Frame(base, width="400", height="500")

#f1.grid(row=0, column=0)
#f2.grid(row=0, column=0)

r = sr.Recognizer()
```

```
def speak(a):
    a.insert(0," ")
    a.delete(0, END)
    MyText=""
    while(MyText==""):
        try:
            with sr.Microphone() as source2:
                r.adjust_for_ambient_noise(source2, duration=0.2)
                audio2 = r.listen(source2)
                MyText = r.recognize_google(audio2)
                a.insert(0, MyText)
        except sr.RequestError as e:
            print("Could not request results; {0}".format(e))
            return
        except sr.UnknownValueError:
            print("unknown error occured")
            return


def swap(frame1, frame2):
    #frame1.grid_forget()
    #frame2.grid(row=0, column=0)
    frame1.pack_forget()
    frame2.pack()
    frame2.tkraise()

def sub_crop(frame):
    global crop_df
    try:
        ans1['text'] = 'Predicted Crop : '
        l=[]
        l.append(float(e1.get()))
        l.append(float(e2.get()))
        l.append(float(e3.get()))
        l.append(float(e4.get()))
        features = crop_df.iloc[:,0:4]
        features = features.append(pd.DataFrame([l],columns=['temperature','humic
        features = scaler2.fit_transform(features)
        gui_feature = features[-1]
        gui_feature = np.reshape(gui_feature,(1,-1))
        lt = model_crop.predict(gui_feature)
        s = 'rice'
        for i in range(0,31):
            if(lt[0][i]==1):
                s = crops[i]
                break
        ans1['text'] = ans1['text'] + s
    except ValueError:
        ans1['text'] = 'ERROR : Invalid Input'




def sub_yield(frame):
    global yield_df
    i=1
    try:
        ans2['text'] = 'Predicted yield: '
        l=[]
        l.append(e5.get()[0].upper() + e5.get()[1:])
        l.append(e6.get()[0].upper() + e6.get()[1:])
        l.append(0.0)
        l.append(float(e7.get()))
        l.append(float(e8.get()))
        l.append(float(e9.get()))
        l.append(float(e10.get()))
        l.append(float(e11.get()))
        l.append(float(e12.get()))
        l.append(float(e13.get()))
        if l[0] not in country_list:
            ans2['text'] = 'Country not supported'
            return
        if l[1] not in item_list:
            ans2['text'] = 'Crop not supported'
            return
        yield_df = yield_df.append(pd.DataFrame([l], columns = ['Area','Item','hg
        yield_df_onehot = pd.get_dummies(yield_df, columns=['Area',"Item"], prefi
        features=yield_df_onehot.loc[:, yield_df_onehot.columns != 'hg/ha_yield']
        label=yield_df['hg/ha_yield']
        features=scaler1.fit_transform(features)
        gui_feature = features[-1]
        gui_feature = np.reshape(gui_feature, (1,-1))
        text = model_yield.predict(gui_feature)
        ans2['text'] = ans2['text'] + str(text[0]) + ' hg/ha'
    except (ValueError, IndexError):
        ans2['text'] = 'ERROR : Invalid Input'
```

In [12]:

```
lab1 = Label(f1, text="PREDICT NOW!!!", font=tkfont.Font(size=25), bg="#81d1e7",
lab1.pack()
lab1.place(x=0, y=50)

but1 = Button(f1, text='CROP', height="5", width="10", font=tkfont.Font(size=15),
but1.pack()
but1.place(x=130, y=120)

but2 = Button(f1, text='YIELD', height="5", width="10", font=tkfont.Font(size=15)
but2.pack()
but2.place(x=130,y=270)
```

In [13]:

```
l = Label(f2, text = "CROP PREDICTION", font=tkfont.Font(size=15), bg="#81d1e7",
l.pack(fill="both", expand=True)
l.place(x=0, y=20)


l1 = Label(f2, text = "Temperature(C)", font=tkfont.Font(size=12), bg="#81d1e7",
l1.pack(fill="both", expand=True)
l1.place(x=30, y=60)

l2 = Label(f2, text = "Humidity(% g/kg)", font=tkfont.Font(size=12), bg="#81d1e7
l2.pack(fill="both", expand=True)
l2.place(x=30, y=100)

l3 = Label(f2, text = "pH", font=tkfont.Font(size=12), bg="#81d1e7", height='2',
l3.pack(fill="both", expand=True)
l3.place(x=30, y=140)

l4 = Label(f2, text = "Average rainfall\nper year(mm)", font=tkfont.Font(size=12
l4.pack(fill="both", expand=True)
l4.place(x=30, y=180)

ans1 = Label(f2, text = "Predicted Crop : ", font=tkfont.Font(size=14), bg="#81d1
ans1.pack(fill="both", expand=True)
ans1.place(x=30, y=290)

e1 = Entry(f2, font=tkfont.Font(size=12), width=15)
e1.pack()
e1.place(x=180, y=68)

e2 = Entry(f2, font=tkfont.Font(size=12), width=15)
e2.pack(fill="both", expand=True)
e2.place(x=180, y=108)

e3 = Entry(f2, font=tkfont.Font(size=12), width=15)
e3.pack()
e3.place(x=180, y=148)

e4 = Entry(f2, font=tkfont.Font(size=12), width=15)
e4.pack(fill="both", expand=True)
e4.place(x=180, y=188)

photo = ImageTk.PhotoImage(Image.open("pic20.png").resize((17,17)))

b1 = Button(f2, command=lambda : speak(e1), image=photo)
b1.pack()
b1.place(x=330, y=68)

b2 = Button(f2, command=lambda : speak(e2), image=photo)
b2.pack()
b2.place(x=330, y=108)

b3 = Button(f2, command=lambda : speak(e3), image=photo)
b3.pack()
b3.place(x=330, y=148)

b4 = Button(f2, command=lambda : speak(e4), image=photo)
```

```
b4.pack()
b4.place(x=330, y=188)

but3 = Button(f2, text='SUBMIT', height="1", width="10", font=tkfont.Font(size=15
but3.pack()
but3.place(x=70,y=240)

but4 = Button(f2, text='BACK', height="1", width="10", font=tkfont.Font(size=15),
but4.pack()
but4.place(x=210,y=240)
```

In [14]:

```
l5 = Label(f3, text = "CROP YIELD PREDICTION", font=tkfont.Font(size=15), bg="#81
l5.pack(fill="both", expand=True)
l5.place(x=0, y=20)


l6 = Label(f3, text = "Country", font=tkfont.Font(size=10), bg="#81d1e7", height=
l6.pack(fill="both", expand=True)
l6.place(x=20, y=60)

l7 = Label(f3, text = "Crop", font=tkfont.Font(size=10), bg="#81d1e7", height='2'
l7.pack(fill="both", expand=True)
l7.place(x=20, y=90)

l8 = Label(f3, text = "Area harvested(ha)", font=tkfont.Font(size=10), bg="#81d1e
l8.pack(fill="both", expand=True)
l8.place(x=20, y=120)

l9 = Label(f3, text = "Average rainfall per year(mm)", font=tkfont.Font(size=10),
l9.pack(fill="both", expand=True)
l9.place(x=20, y=150)

l10 = Label(f3, text = "Pesticides (kg/ha)", font=tkfont.Font(size=10), bg="#81d1
l10.pack(fill="both", expand=True)
l10.place(x=20, y=180)

l11 = Label(f3, text = "Average Temperature(C)", font=tkfont.Font(size=10), bg="#
l11.pack(fill="both", expand=True)
l11.place(x=20, y=210)

l12 = Label(f3, text = "Nitrogen fertilizer (kg/ha)", font=tkfont.Font(size=10),
l12.pack(fill="both", expand=True)
l12.place(x=20, y=240)

l13 = Label(f3, text = "Phosphorous fertilizer (kg/ha)", font=tkfont.Font(size=10
l13.pack(fill="both", expand=True)
l13.place(x=20, y=270)

l14 = Label(f3, text = "Potassium fertilizer (kg/ha)", font=tkfont.Font(size=10),
l14.pack(fill="both", expand=True)
l14.place(x=20, y=300)

ans2 = Label(f3, text = "Predicted yield: ", font=tkfont.Font(size=14), bg="#81d1
ans2.pack(fill="both", expand=True)
ans2.place(x=20, y=390)

e5 = Entry(f3, font=tkfont.Font(size=10), width=15)
e5.pack()
e5.place(x=200, y=65)

e6 = Entry(f3, font=tkfont.Font(size=10), width=15)
e6.pack(fill="both", expand=True)
e6.place(x=200, y=95)

e7 = Entry(f3, font=tkfont.Font(size=10), width=15)
e7.pack()
e7.place(x=200, y=125)
```

```
e8 = Entry(f3, font=tkfont.Font(size=10), width=15)
e8.pack(fill="both", expand=True)
e8.place(x=200, y=155)

e9 = Entry(f3, font=tkfont.Font(size=10), width=15)
e9.pack(fill="both", expand=True)
e9.place(x=200, y=185)

e10 = Entry(f3, font=tkfont.Font(size=10), width=15)
e10.pack()
e10.place(x=200, y=215)

e11 = Entry(f3, font=tkfont.Font(size=10), width=15)
e11.pack(fill="both", expand=True)
e11.place(x=200, y=245)

e12 = Entry(f3, font=tkfont.Font(size=10), width=15)
e12.pack()
e12.place(x=200, y=275)

e13 = Entry(f3, font=tkfont.Font(size=10), width=15)
e13.pack(fill="both", expand=True)
e13.place(x=200, y=305)

b5 = Button(f3, command=lambda : speak(e5), image=photo)
b5.pack()
b5.place(x=322, y=65)

b6 = Button(f3, command=lambda : speak(e6), image=photo)
b6.pack()
b6.place(x=322, y=95)

b7 = Button(f3, command=lambda : speak(e7), image=photo)
b7.pack()
b7.place(x=322, y=125)

b8 = Button(f3, command=lambda : speak(e8), image=photo)
b8.pack()
b8.place(x=322, y=155)

b9 = Button(f3, command=lambda : speak(e9), image=photo)
b9.pack()
b9.place(x=322, y=185)

b10 = Button(f3, command=lambda : speak(e10), image=photo)
b10.pack()
b10.place(x=322, y=215)

b11 = Button(f3, command=lambda : speak(e11), image=photo)
b11.pack()
b11.place(x=322, y=245)

b12 = Button(f3, command=lambda : speak(e12), image=photo)
b12.pack()
b12.place(x=322, y=275)
```

```
b13 = Button(f3, command=lambda : speak(e13), image=photo)
b13.pack()
b13.place(x=322, y=305)

but5 = Button(f3, text='SUBMIT', height="1", width="10", font=tkfont.Font(size=15
but5.pack()
but5.place(x=70,y=345)

but6 = Button(f3, text='BACK', height="1", width="10", font=tkfont.Font(size=15),
but6.pack()
but6.place(x=210,y=345)
```

In [15]:

```
f1.pack()
f1.tkraise()

base.wm_attributes("-transparentcolor","grey")
base.title("Crop and yield prediction window")
base.mainloop()
```

In [ ]: