from cloudant.client import Cloudant

client = Cloudant.iam('6f4f5183-072e-4bd0-b33f-8c0562a8e227-bluemix', 'IcNwIUOYQsMwH32\_e2m3xg93E-Af0KVHeiAwVijUAWWC', connect=True)

my\_database=client['my\_database']

import numpy as np

import os

from flask import Flask, app,request,render\_template,redirect,url\_for,session

from tensorflow.keras import models

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

from tensorflow.python.ops.gen\_array\_ops import concat

from tensorflow.keras.applications.inception\_v3 import preprocess\_input

import requests

os.add\_dll\_directory

model1=load\_model(r'D:\Usman\IBM Project\Model\body.h5')

model2=load\_model(r'D:\Usman\IBM Project\Model\level.h5')

app=Flask(\_\_name\_\_)

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/index.html')

def home():

return render\_template("index.html")

@app.route('/register.html')

def register():

return render\_template("register.html")

@app.route('/afterreg',methods=['POST'])

def afterreg():

x = [x for x in request.form.values()]

print(x)

data={'\_id':x[1],'name':x[0],'psw':x[2]}

print(data)

query={'\_id':{'$eq':data['\_id']}}

docs=my\_database.get\_query\_result(query)

print(docs)

print(len(docs.all()))

if (len(docs.all())==0):

url=my\_database.create\_document(data)

return render\_template("register.html",pred="Registration Successful, please login with your details")

else:

return render\_template("register.html",pred="You are already a member, please login using your registered details")

@app.route('/login.html')

def login():

return render\_template("login.html")

@app.route('/afterlogin',methods=['POST'])

def afterlogin():

user=request.form['\_id']

passw=request.form['psw']

print(user,passw)

query={'\_id':{'$eq':user}}

docs=my\_database.get\_query\_result(query)

print(docs)

print(len(docs.all()))

if (len(docs.all())==0):

return render\_template("login.html",pred="The username or password is incorrect. Please login with correct details.")

else:

if((user==docs[0][0]['\_id']and passw==docs[0][0]['psw'])):

return redirect(url\_for('prediction'))

else:

return render\_template("login.html",pred="The username is not found or the details you've entered is incorrect.")

@app.route('/logout.html')

def logout():

return render\_template("logout.html")

@app.route('/prediction.html')

def prediction():

return render\_template("prediction.html")

@app.route('/result',methods=["GET","POST"])

def result():

if request.method=="POST":

f=request.files['file']

basepath=os.path.dirname("\_\_file\_\_")

filepath=os.path.join(basepath,'uploads', f.filename)

f.save(filepath)

img=image.load\_img(filepath,target\_size=(256, 256))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

img\_data=preprocess\_input(x)

prediction1=np.argmax(model1.predict(img\_data))

prediction2=np.argmax(model2.predict(img\_data))

index1=['front','rear','side']

index2=['minor','moderate','severe']

result1=index1[prediction1]

result2=index2[prediction2]

print(result1)

print(result2)

if(result1=="front"and result2=="minor"):

value="3000 - 5000 INR"

elif(result1=="front"and result2=="moderate"):

value="6000 - 8000 INR"

elif(result1=="front"and result2=="severe"):

value="9000 - 11000 INR"

elif(result1=="rear"and result2=="minor"):

value="4000 - 6000 INR"

elif(result1=="rear"and result2=="moderate"):

value="7000 - 9000 INR"

elif(result1=="rear"and result2=="severe"):

value="11000 - 13000 INR"

elif(result1=="side"and result2=="minor"):

value="6000 - 8000 INR"

elif(result1=="side"and result2=="moderate"):

value="9000 - 11000 INR"

elif(result1=="side"and result2=="severe"):

value="12000 - 15000 INR"

else:

value="16000 - 50000 INR"

return render\_template("result.html", prediction="The Estimated cost for the damage is: "+value)

if \_\_name\_\_=="\_\_main\_\_":

app.run(debug=False,port=8080)