```
!pip install flask-ngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Requirement already satisfied: flask-ngrok in /usr/local/lib/python3.7/dist-packages (0
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: click<8.0,>=5.1 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: Werkzeug<2.0,>=0.15 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: Jinja2<3.0,>=2.10.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: itsdangerous<2.0,>=0.24 in /usr/local/lib/python3.7/dist
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/li
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (
```

```
import os
if not os.path.exists('templates'):
  os.mkdir('templates')
if not os.path.exists('static'):
  os.mkdir('static')
if not os.path.exists('static/input_img'):
  os.mkdir('static/input_img')
if not os.path.exists('static/images'):
  os.mkdir('static/images')
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mou
```

```
!pip install pyngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Requirement already satisfied: pyngrok in /usr/local/lib/python3.7/dist-packages (5.1.0
Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from p
```

```
!pip install --upgrade ibm_db
!pip install --upgrade ibm_db_sa
!pip install --upgrade SQLAlchemy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Collecting ibm_db
  Downloading ibm_db-3.1.3.tar.gz (1.4 MB)
     |████████████████████████████████| 1.4 MB 7.0 MB/s
```

```
    Installing build dependencies ... done
    Getting requirements to build wheel ... done
    Installing backend dependencies ... done
      Preparing wheel metadata ... done
  Building wheels for collected packages: ibm-db
    Building wheel for ibm-db (PEP 517) ... done
    Created wheel for ibm-db: filename=ibm_db-3.1.3-cp37-cp37m-linux_x86_64.whl size=4146
    Stored in directory: /root/.cache/pip/wheels/a7/fe/6f/52ae8e5a30a0626cec5f28f908e4d2c
  Successfully built ibm-db
  Installing collected packages: ibm-db
  Successfully installed ibm-db-3.1.3
  Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
  Collecting ibm_db_sa
    Downloading ibm_db_sa-0.3.8-py3-none-any.whl (30 kB)
  Requirement already satisfied: ibm-db>=2.0.0 in /usr/local/lib/python3.7/dist-packages
  Requirement already satisfied: sqlalchemy>=0.7.3 in /usr/local/lib/python3.7/dist-packa
  Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-pack
  Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.7/dist-packag
  Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dis
  Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (fro
  Installing collected packages: ibm-db-sa
  Successfully installed ibm-db-sa-0.3.8
  Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
  Requirement already satisfied: SQLAlchemy in /usr/local/lib/python3.7/dist-packages (1.
  Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.7/dist-packag
  Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-pack
  Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (fro
  Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dis
```

```
import numpy as np
import keras
from keras.models import load_model
from keras.preprocessing import image


import tensorflow as tf


UPLOAD_FOLDER='/content/static/input_img'



foodlist=['Apple',
 'Badam',
 'Badam Drink',
 'Banana',
 'Beef Steak',
 'Beetroot Fry',
 'Biriyani',
 'Biscuits',
 'Bitter Guard Fry',
 'Boiled egg',
 'Bread and Jam',
 'Bread with Peanutbutter',
 'Burger',
 'Capsicum Curry',
 'Cashew',
 'Cauliflower Fry',
 'Chappathi',
 'Cheeseballs',
 'Chilli Beef',
 'Chocolate',
 'Chocolate Icecream',
 'Choolapoori with Channa',
 'Coffee or Latte',
 'Crab Masala',
 'Cucumber',
 'Curdrice',
 'Dosa',
 'Dragon Fruit',
 'Drumstick Gravy',
 'Dry Grapes',
 'Falooda',
 'Fig',
 'Fish Fry',
 'French Fries',
 'Fresh Juices',
 'Fried Rice',
 'Grapes',
 'Grill Chicken',
 'Gulab Jamun',
 'Halwa',
 'Ice Apple',
 'Idiyappam',
 'Idly',
 'Ivygourd Fry',
 'Jilebi',
 'Ladys Finger',
 'Lemon Rice',
 'Maa Ladoo',
 'Mango',
 'Milk',
 'Momos',
 'Murukku',
 'Mushroom Gravy',
 'Nachos',
 'Oats',
 'Omelette',
 'Orange',
 'Panner Butter Masala',
 'Parota',
 'Pasta',
 'Pineapple',
 'Pistas',
 'Pizza',
 'Plain Bread',
 'Pongal',
 'Poori',
 'Pork Bbq',
 'Potato Fry',
 'Prawn',
 'Puttu',
 'Rasagulla',
 'Rasamalai',
 'Ravadosa',
 'Rose Milk',
 'Sambar Idly',
 'Sambar Vada',
 'Samosa',
 'Sapota',
 'Shawarma',
 'Soft Drinks',
 'Spinach Gravy',
 'Springroll',
 'Sprouts',
 'Steamed Redrice',
 'Tamarind Rice',
 'Tea',
 'Tender Coconut',
 'Vadapav',
 'Vanilla Icecream',
 'Wheat Bread',
 'Whiterice with Spinach',
 'Whiterice with Vegetablestew']
```

```
from werkzeug.utils import secure_filename
from IPython.core.profiledir import ProfileDirError
from flask_ngrok import run_with_ngrok
from flask import Flask,render_template,request,redirect,url_for,session
from pyngrok import ngrok
ngrok.set_auth_token("2Ezqku6JC7CLgsXAuubEgbp62j4_BAoQouoxu41Cthe1m834")
import ibm_db
import json
import requests
import pickle
import os
import io
Weight=0
Height=0
model=pickle.load(open('/content/drive/MyDrive/IBM_HACK CHALLENGE 2022/bmi.pkl','rb'))
calorie_model=keras.models.load_model('/content/drive/MyDrive/IBM_HACK CHALLENGE 2022/food_pr
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l
connState=ibm_db.active(conn)
```

```
    print(connState)
    app=Flask(__name__)
    run_with_ngrok(app)
    @app.route('/')
    def login():
      return render_template('login.html')

    @app.route('/welcome',methods=["GET"])
    def welcome():
      if request.method=="GET":
        email=request.args.get('Email')
        password=request.args.get('Password')
        print(email,"\n",password)
        login="select * from REGISTER where EMAIL=email and PASSWORD=password"
        stmt=ibm_db.prepare(conn,login)
        ibm_db.execute(stmt)
      return render_template('home.html')

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

    def register():

      return render_template('register.html')

    @app.route('/success',methods=["GET"])
    def success():
      if request.method=="GET":
        name=request.args.get('Name')
        email=request.args.get('Email')
        password=request.args.get('Password')
        phone=request.args.get('phone')
        print(name,"\n",email,"\n",password,"\n",phone)
        insert_sql="INSERT INTO REGISTER VALUES (?,?,?,?)"
        stmt=ibm_db.prepare(conn,insert_sql)
        ibm_db.bind_param(stmt,1,name)
        ibm_db.bind_param(stmt,2,email)
        ibm_db.bind_param(stmt,3,password)
        ibm_db.bind_param(stmt,4,phone)
        ibm_db.execute(stmt)
      return render_template('login.html')

    @app.route('/bmicalculator',methods=["POST"])
    def bmicalculator():
      return render_template("bmi.html")

    @app.route('/bmi',methods=["GET"])
    def calculate():
      if request.method=="GET":
        weight=request.args.get('Weight')
        height=request.args.get('Height')
        arr=model.predict([[weight,height]])
        t=float(arr)
      print(t)
      return render_template("bmi.html",info=t)

  @app.route('/calorie',methods=["POST"])
  def calorie():
    return render_template('calories.html')

  @app.route('/predictcalories',methods=["POST"])
  def predictcalories():
    if request.method=="POST":
      image=request.files['image']
      filename=(secure_filename(image.filename))
      app.secret_key = "secret key"
      app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER
      image.save(os.path.join(app.config['UPLOAD_FOLDER'],filename))
      print(image)
      img = tf.keras.utils.load_img('/content/static/input_img/'+filename,target_size=(64,64))
      x=tf.keras.utils.img_to_array(img)
      x=np.expand_dims(x,axis=0)
      pred=np.argmax(calorie_model.predict(x))
      prediction=foodlist[pred]
      calories="select CALORIES from CALORIES_TABLE where FOOD_NAME= "+"\'"+prediction+"\'"
      st=ibm_db.exec_immediate(conn,calories)
      while(ibm_db.fetch_row(st)!=False) :
        predict_cal=ibm_db.result(st,0)
      print(predict_cal)
      cal=request.form.get('Quantity')
      print(cal)
      calorie_num=float(predict_cal)
      q_num=float(cal)
      result_cal=float(calorie_num*q_num)
      n=str(result_cal)
      return render_template("calories.html",info="FOOD:"+prediction,calories="Calories:"+n)

  @app.route('/biologicalage',methods=["POST"])
  def biologicalage():

      return render_template('biologicalage.html')

  @app.route('/bioagecalc',methods=["GET"])
  def bioagecalc():
    if request.method=="GET":
      age=float(request.args.get('Age'))
      weight=float(request.args.get('Weight'))
      height=float(request.args.get('Height'))
      arr=model.predict([[weight,height]])
      t=float(arr)
      if age<18.5:
        t=t+1
      elif age >=18.5 and age<=25:
        t=t-1
      elif age>25 and age<=29.9:
        t=t+2
      else:
        t=t+3
      return render_template('biologicalage.html',info=t)


  if __name__=='__main__':
      app.run()
```

```
True
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Running on http://7a23-104-198-4-138.ngrok.io
 * Traffic stats available on http://127.0.0.1:4040
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:25:44] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:25:44] "GET /static/style.css HTTP/1.1" 200
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:25:45] "GET /static/images/bg_2.jpg HTTP/1.
raji@gmail.com
 raj@12
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:25:56] "GET /welcome?Email=raji%40gmail.com
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:25:56] "GET /favicon.ico HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:26:01] "POST /bmicalculator HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:26:01] "GET /static/images/bmi_img.jpg HTTP
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:566: FutureWarning: Arrays of by
  X = check_array(X, **check_params)
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:26:10] "GET /bmi?Weight=50&Height=150 HTTP/
22.788611063838218
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:26:31] "POST /calorie HTTP/1.1" 200 -
<FileStorage: 'th - 2022-09-22T204026.775.jpg' ('image/jpeg')>
1/1 [==============================] - 0s 112ms/step
214
1
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:27:15] "POST /predictcalories HTTP/1.1" 200
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:27:33] "POST /biologicalage HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Oct/2022 14:27:54] "GET /bioagecalc?Age=45&Weight=50&He
```

✓ 3m 51s completed at 19:59