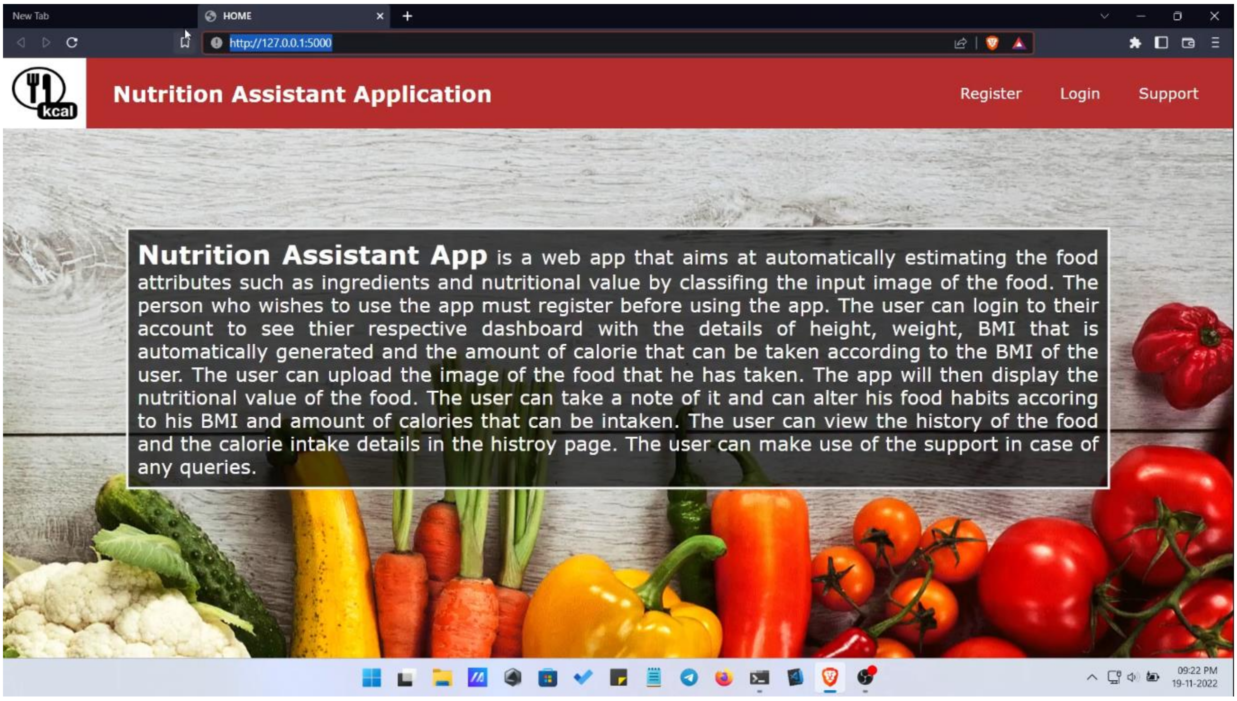
Title	Implementing web application
Team ID	PNT2022TMID39029
Project Name	Project – Nutrition Assistant Application

Step 1: Creating UI to interact with web application

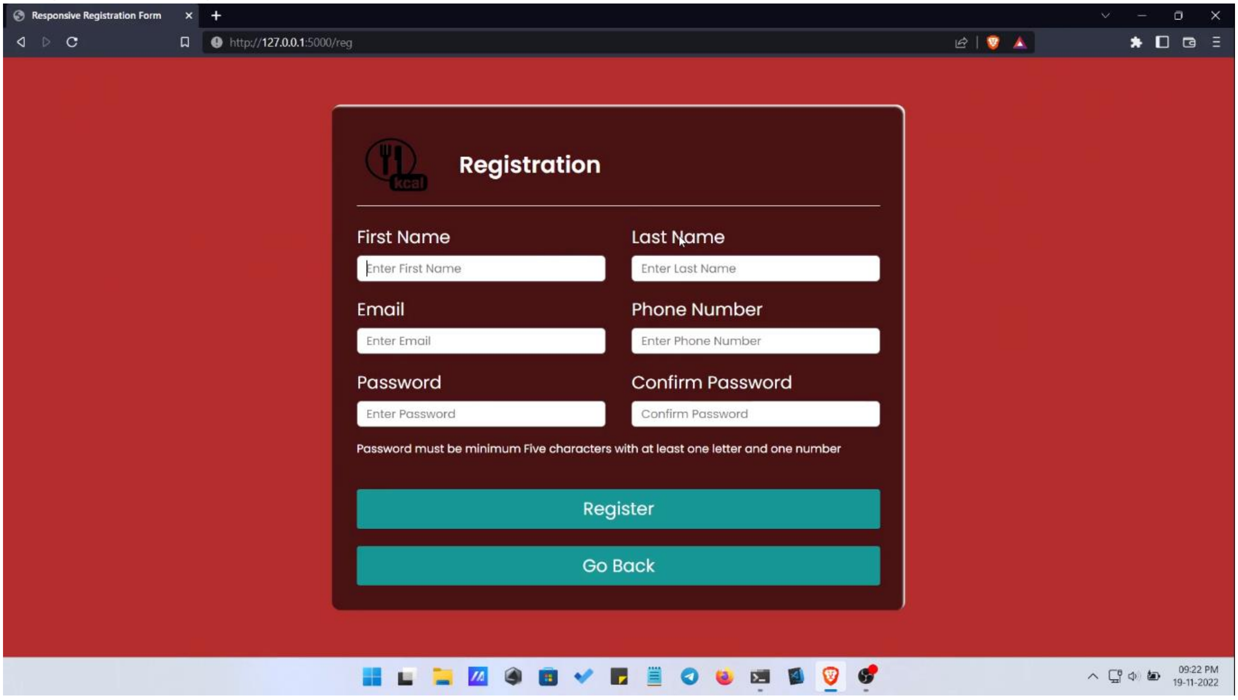
Home Page:



Nutrition Assistant Application

Nutrition Assistant App is a web app that aims at automatically estimating the food attributes such as ingredients and nutritional value by classifing the input image of the food. The person who wishes to use the app must register before using the app. The user can login to their account to see thier respective dashboard with the details of height, weight, BMI that is automatically generated and the amount of calorie that can be taken according to the BMI of the user. The user can upload the image of the food that he has taken. The app will then display the nutritional value of the food. The user can take a note of it and can alter his food habits accoring to his BMI and amount of calories that can be intaken. The user can view the history of the food and the calorie intake details in the histroy page. The user can make use of the support in case of any queries.

Registration Page:



Registration

First Name: Enter First Name

Last Name: Enter Last Name

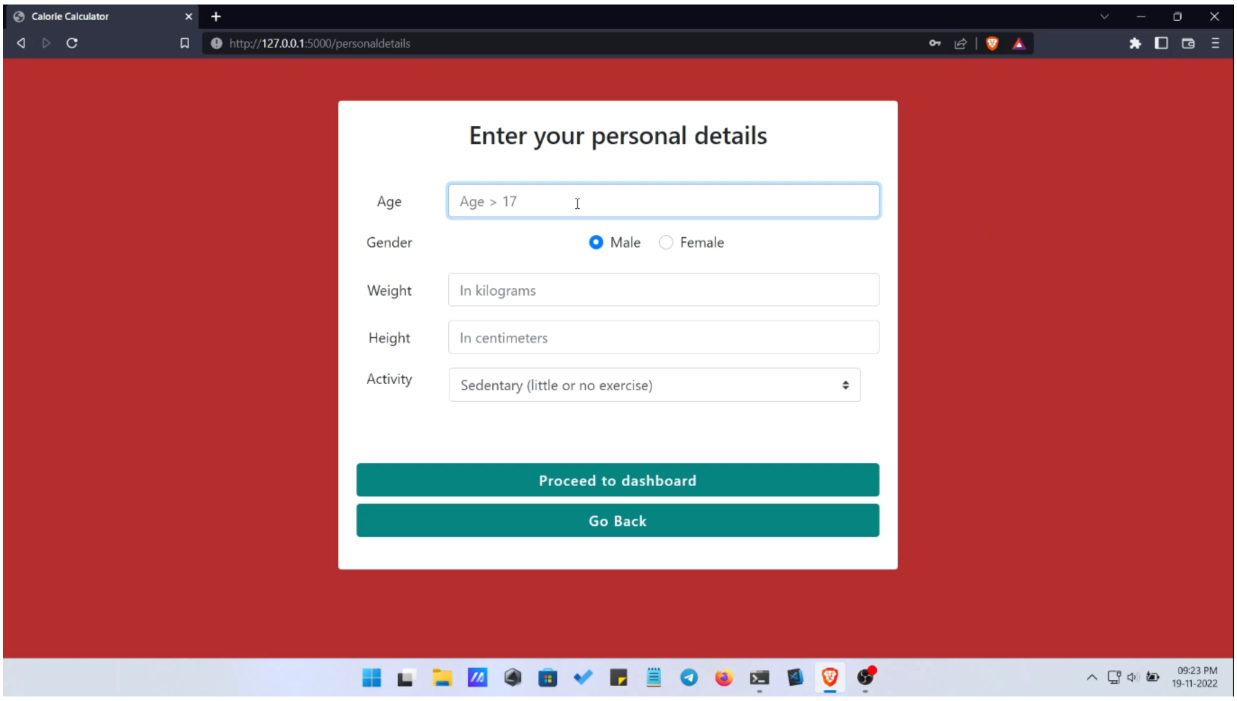
Email: Enter Email

Phone Number: Enter Phone Number

Password: Enter Password

Confirm Password: Confirm Password

Password must be minimum Five characters with at least one letter and one number



Enter your personal details

Age: Age > 17

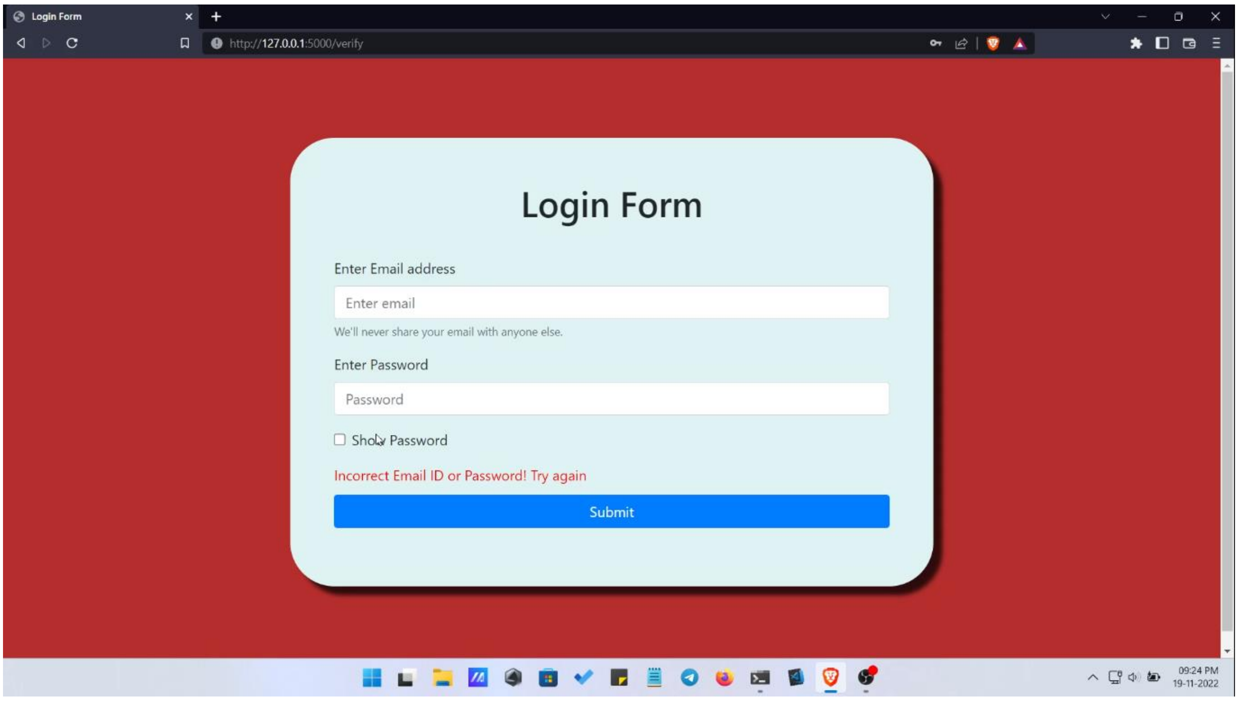
Gender: ● Male ○ Female

Weight: In kilograms

Height: In centimeters

Activity: Sedentary (little or no exercise)

Login Page:



Login Form

Enter Email address

Enter email

We'll never share your email with anyone else.

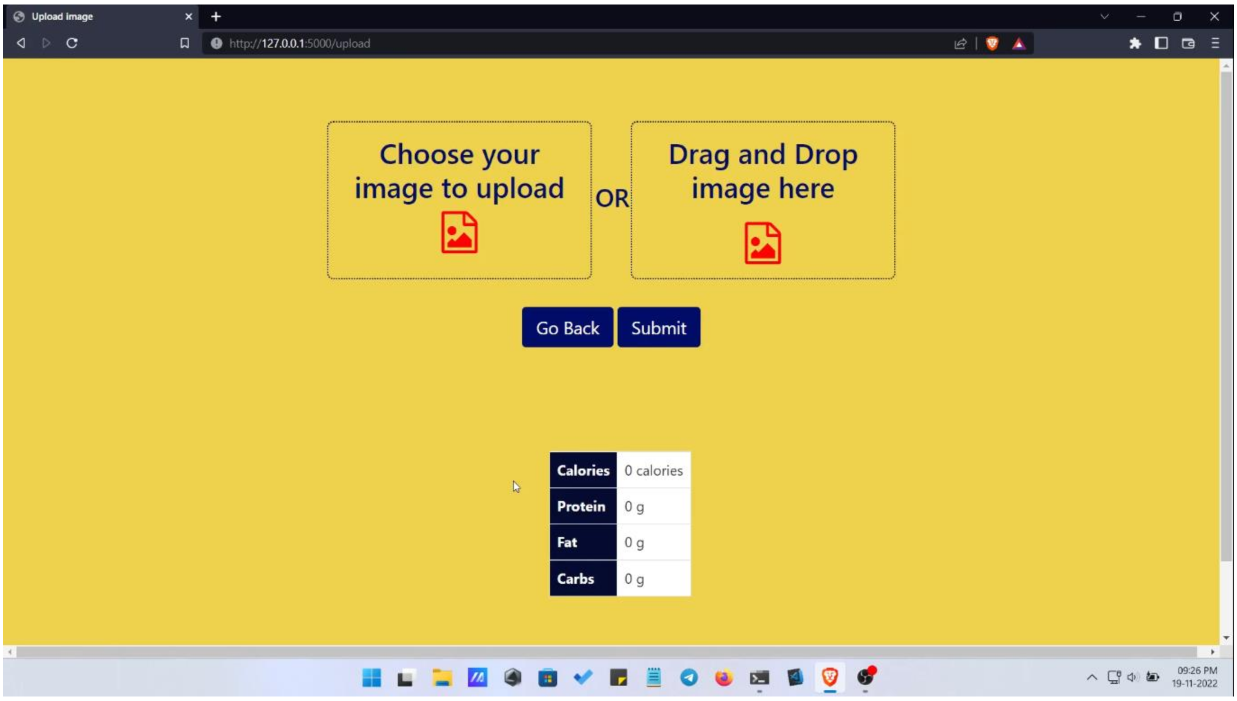
Enter Password

Password

□ Show Password

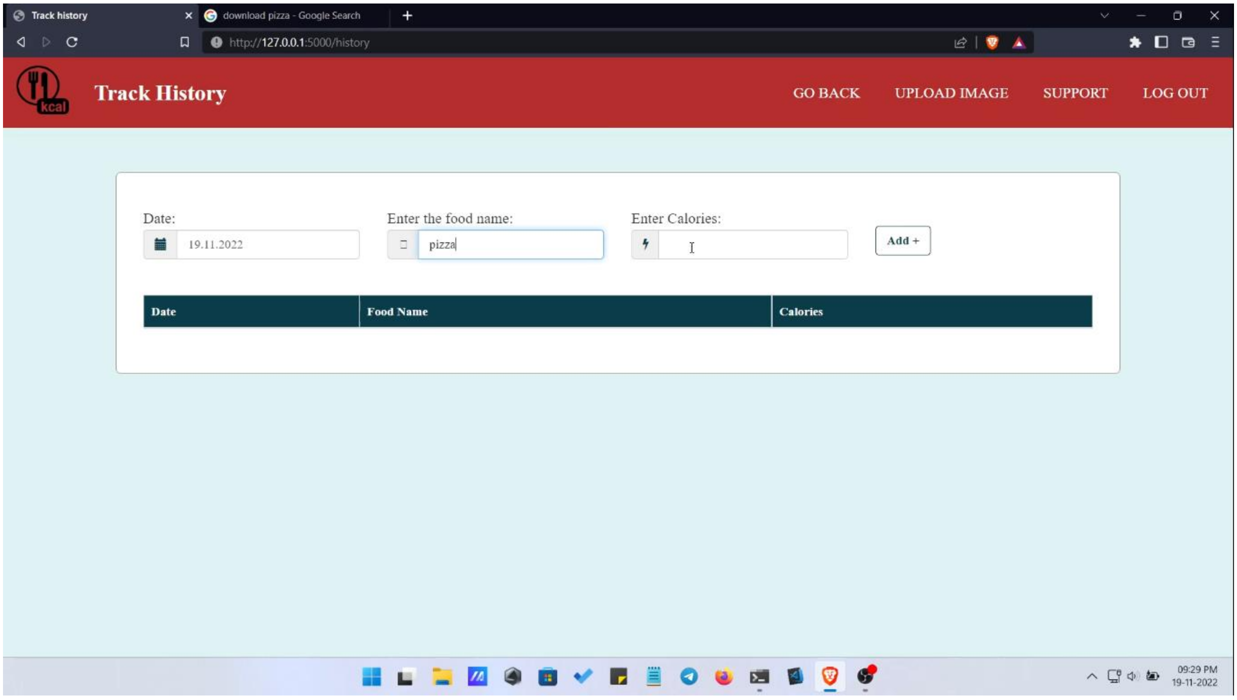
Incorrect Email ID or Password! Try again

Upload Image:



Calories	0 calories
Protein	0 g
Fat	0 g
Carbs	0 g

Track history:



Date: 19.11.2022

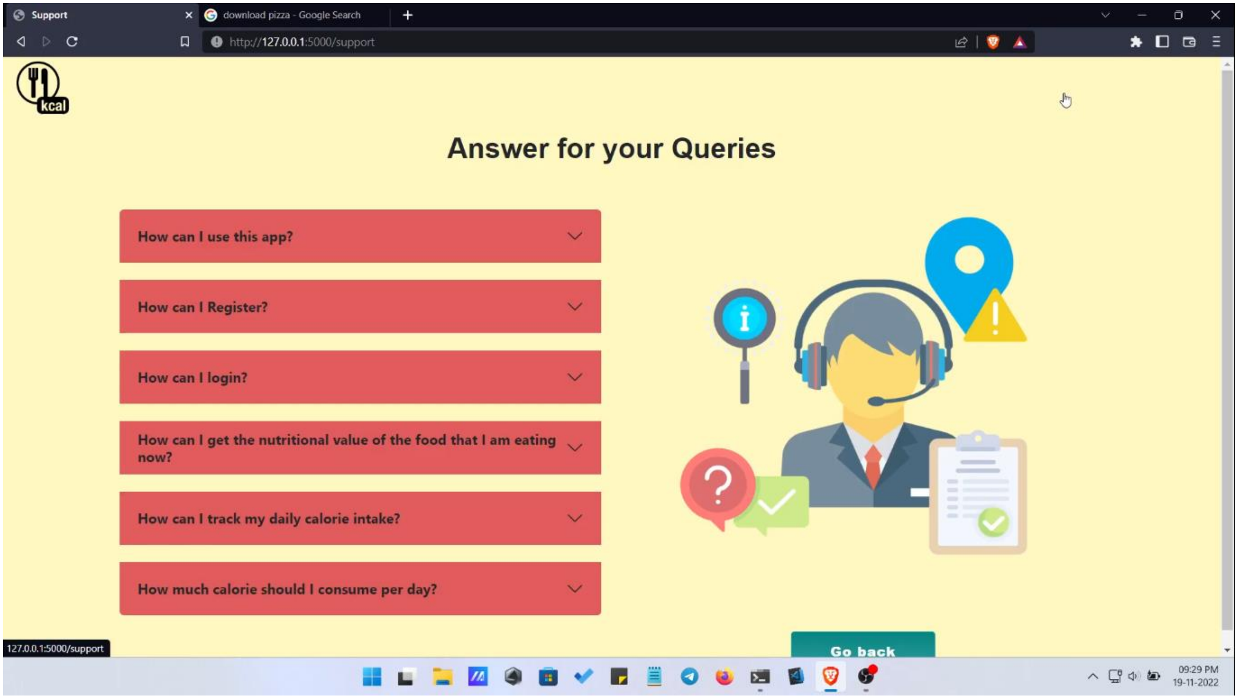
Enter the food name: pizza

Enter Calories:

Add +

Date	Food Name	Calories

Support Page:



Answer for your Queries

How can I use this app?

How can I Register?

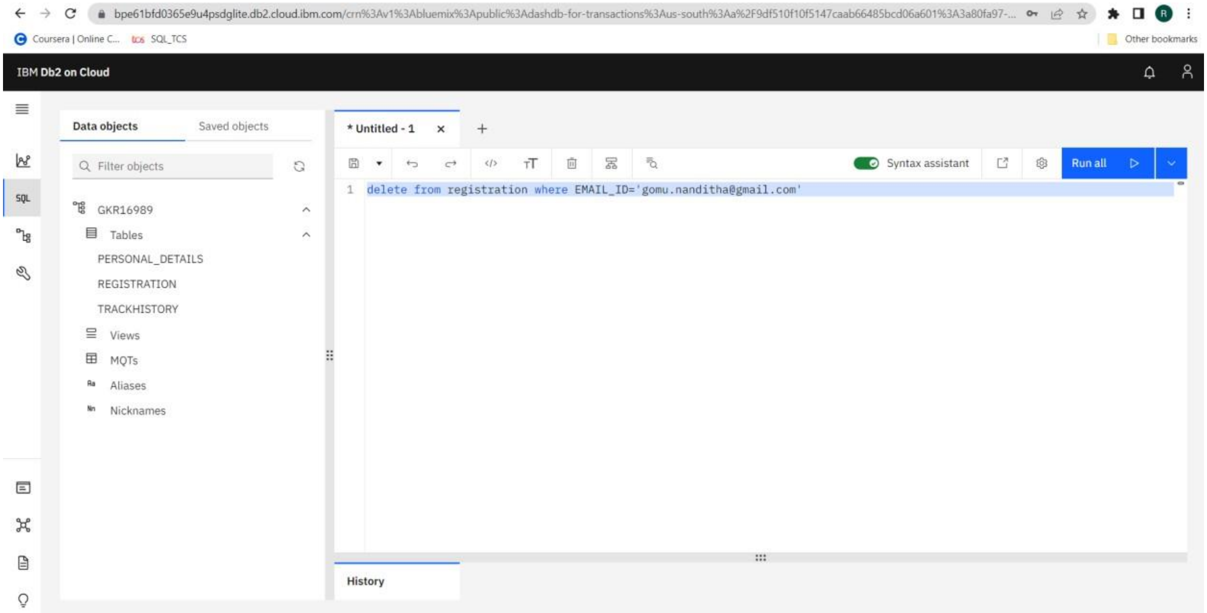
How can I login?

How can I get the nutritional value of the food that I am eating now?

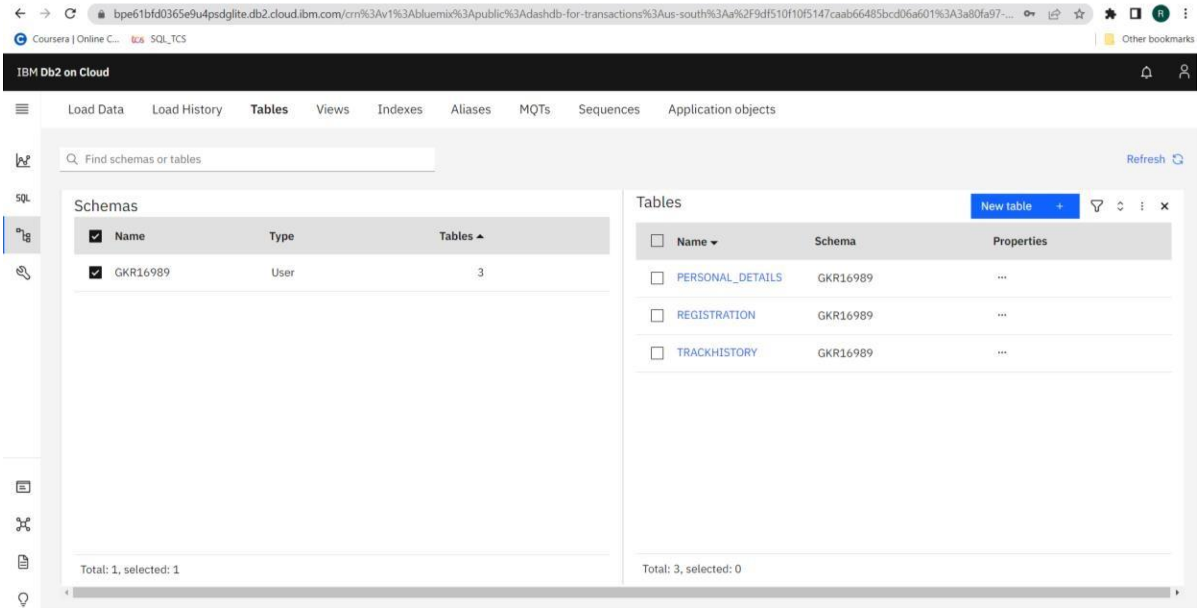
How can I track my daily calorie intake?

How much calorie should I consume per day?

Step 2: Create IBM DB2 and connect with python



```sql
delete from registration where EMAIL_ID='gomu.nanditha@gmail.com'
```



Schemas

	Name	Type	Tables ▲
☑	GKR16989	User	3

Total: 1, selected: 1

Tables

	Name ▾	Schema	Properties
□	PERSONAL_DETAILS	GKR16989	···
□	REGISTRATION	GKR16989	···
□	TRACKHISTORY	GKR16989	···

Total: 3, selected: 0

Connection with python

```
app=Flask(_name_)

app.secret_key='a'

try:

    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30120;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gkr16989;PWD=WvN7xr79Kp6YfdL7","","")

except:

    print("Unable to connect: ",ibm_db.conn_error())
```

Step 3: API connectio code python

```
from clarifai_grpc.grpc.api import service_pb2,resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc

stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())


YOUR_CLARIFAI_API_KEY="064920064d594a4fad0c07262b246a75"
YOUR_APPLICATION_ID="Nutrition Assistant Application"
img = request.files['file']
        print("working")

        path='./static/'+session['email']+'.jpg'
        img.save(path)

        metadata=(('authorization','Key '+YOUR_CLARIFAI_API_KEY),)
        with open(path,"rb") as f:
          file_bytes=f.read()
          request1=service_pb2.PostModelOutputsRequest(
            model_id='9504135848be0dd2c39bdab0002f78e9',
            inputs=[
              resources_pb2.Input(
                data=resources_pb2.Data(
                  image=resources_pb2.Image(
                    base64=file_bytes
                  )
                )
              )
            ])
```

```
      response =stub.PostModelOutputs(request1, metadata=metadata)
      if response.status.code != status_code_pb2.SUCCESS:
        raise Exception("Request failed, status code: " + str(response.status.code))


      for concept in response.outputs[0].data.concepts:
        print('%12s: %.2f' % (concept.name, concept.value))
      api_url = 'https://api.spoonacular.com/recipes/guessNutrition?title='
      query = response.outputs[0].data.concepts[0].name
      response = requests.get(api_url + query, headers={'X-Api-Key':
'8f123f2f983b4b69bfe1e4a25f7bfb06'})
      if response.status_code == requests.codes.ok:
        fullresponse=response.json()
        calories=str(fullresponse['calories']['value'])+' '+str(fullresponse['calories']['unit'])
        protein=str(fullresponse['protein']['value'])+' '+str(fullresponse['protein']['unit'])
        fat=str(fullresponse['fat']['value'])+' '+str(fullresponse['fat']['unit'])
        carbs=str(fullresponse['carbs']['value'])+' '+str(fullresponse['carbs']['unit'])
        print(calories,protein,fat,carbs)
        print(type(fullresponse['calories']['value']))
        print(fullresponse)
      else:
        print("Error:", response.status_code, response.text)
      return render_template('upload.html',calories=calories,fat=fat,protein=protein,carbs=carbs)
    except Exception as e:
      print(e)
    return "Error Occured"
```