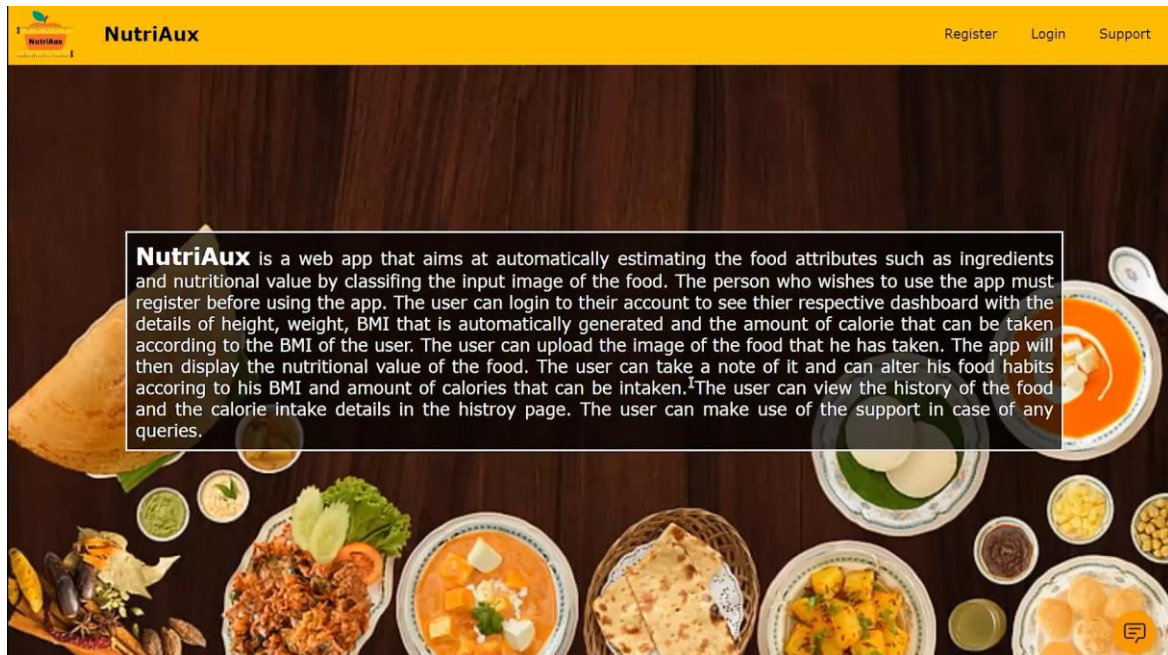


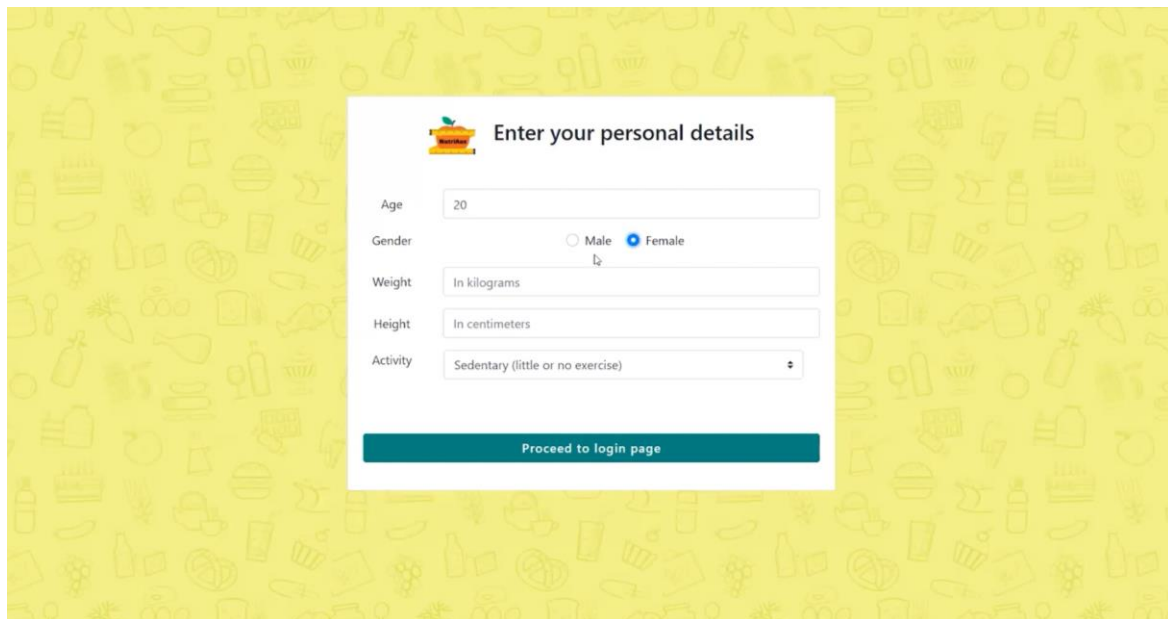
|                     |   |
|---------------------|---|
| <b>Title</b>        | Implementing web application              |
| <b>Team ID</b>      | PNT2022TMID27987                          |
| <b>Project Name</b> | Project – Nutrition Assistant Application |


## Step 1: Creating UI to interact with web application

### Home Page:



### Registration Page:



 **Enter your personal details**

Age

Gender ☐ Male ☒ Female

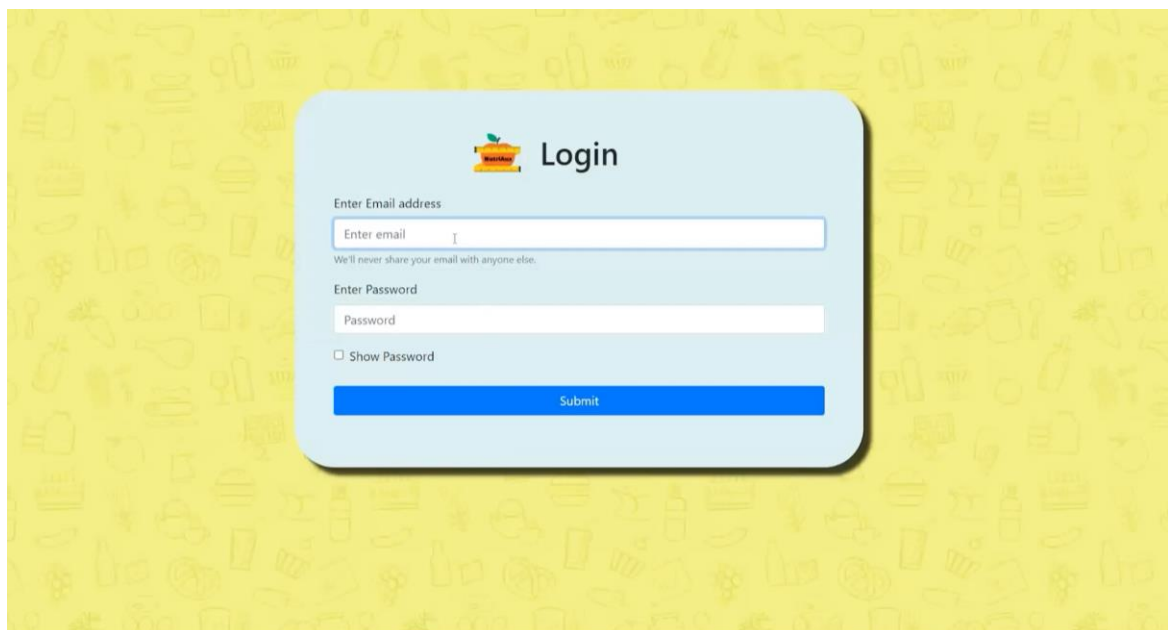
Weight


Height

Activity

[Proceed to login page](#)

**Login Page:**



 **Login**

Enter Email address

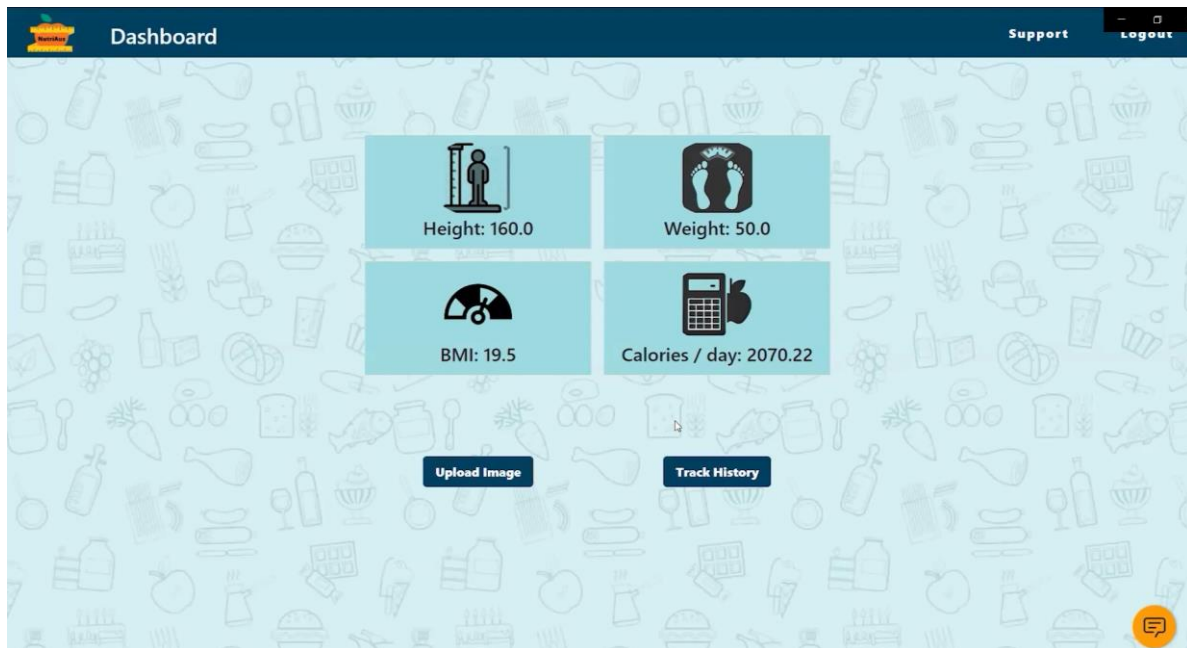
We'll never share your email with anyone else.

Enter Password

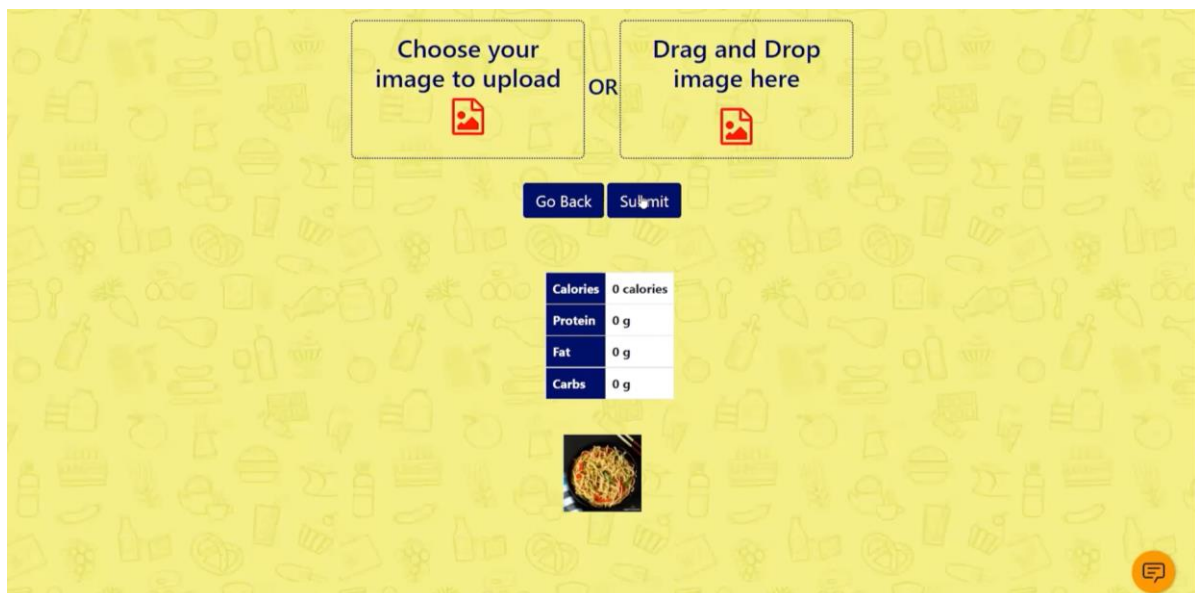
☐ Show Password

[Submit](#)


## Dashboard:




## Upload Image:





## Track history:

 **Track History**

GO BACK    UPLOAD IMAGE    SUPPORT    LOG OUT


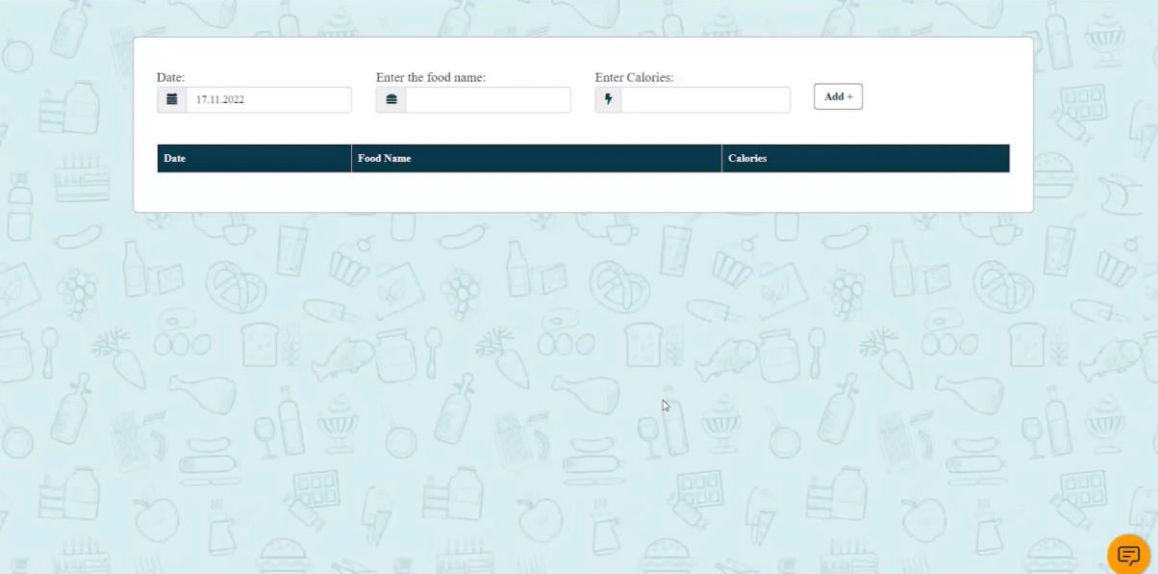
Date:  
 17.11.2022

Enter the food name:  


Enter Calories:  


Add +

| Date | Food Name | Calories |
|------|-----------|----------|
|------|-----------|----------|



## Support Page:

Feeling stuck ? We are here to help you :)

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? ✓

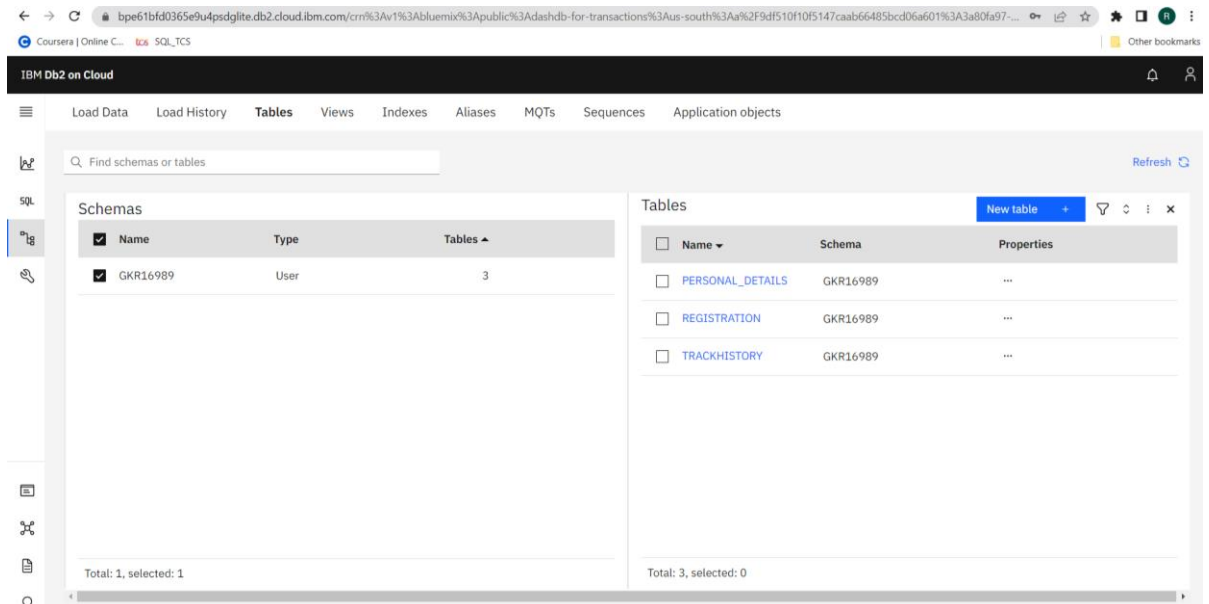
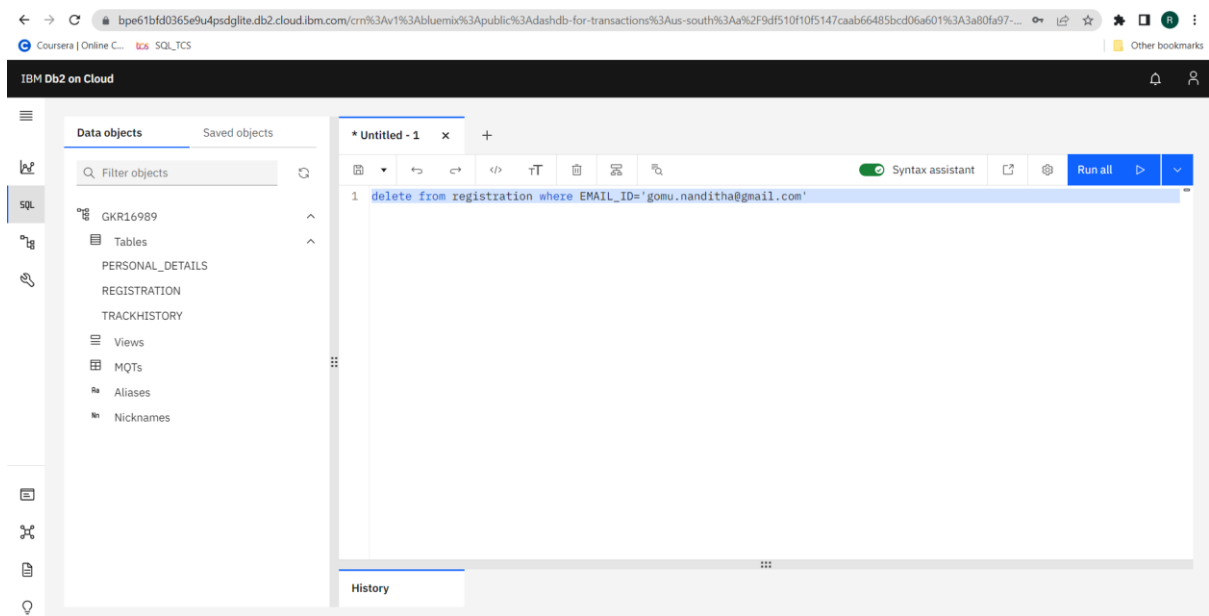
How can I track my daily calorie intake? ✓

How much calorie should I consume per day? ✓



Go back

## Step 2: Create IBM DB2 and connect with python



## Connection with python

```
app=Flask(__name__)
```

```
app.secret_key='a'
```

```
try:
```

```
    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-  
d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;SECURITY=SSL;SSLSer  
verCertificate=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="747efb7b11334fdab65de5ba402969dd"
```

```
YOUR_APPLICATION_ID="nutriaux"
```

```
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"

```