

Implementing Web Application Integrate Nutrition API

Team ID	PNT2022TMID49745
Project Name	Nutrition Assistant Application

Clarifai:

The screenshot displays the Clarifai Community web application interface. The browser address bar shows 'clarifai.com/settings'. The user is logged in as 'Nithisha S'. The left sidebar contains navigation links: Account, Billing, Security, and Usage. The main content area is titled 'Account Settings' and is divided into two sections: 'Contact Information' and 'Profile'.

Contact Information

First Name *	Last Name *
<input type="text" value="Nithisha"/>	<input type="text" value="S"/>
Company *	Country *
<input type="text" value="Dr. Sivanthi Aditanar College Of Engineering"/>	<input type="text" value="India"/>
Job Title *	Job Role *
<input type="text" value="Student"/>	<input type="text" value="Other"/>

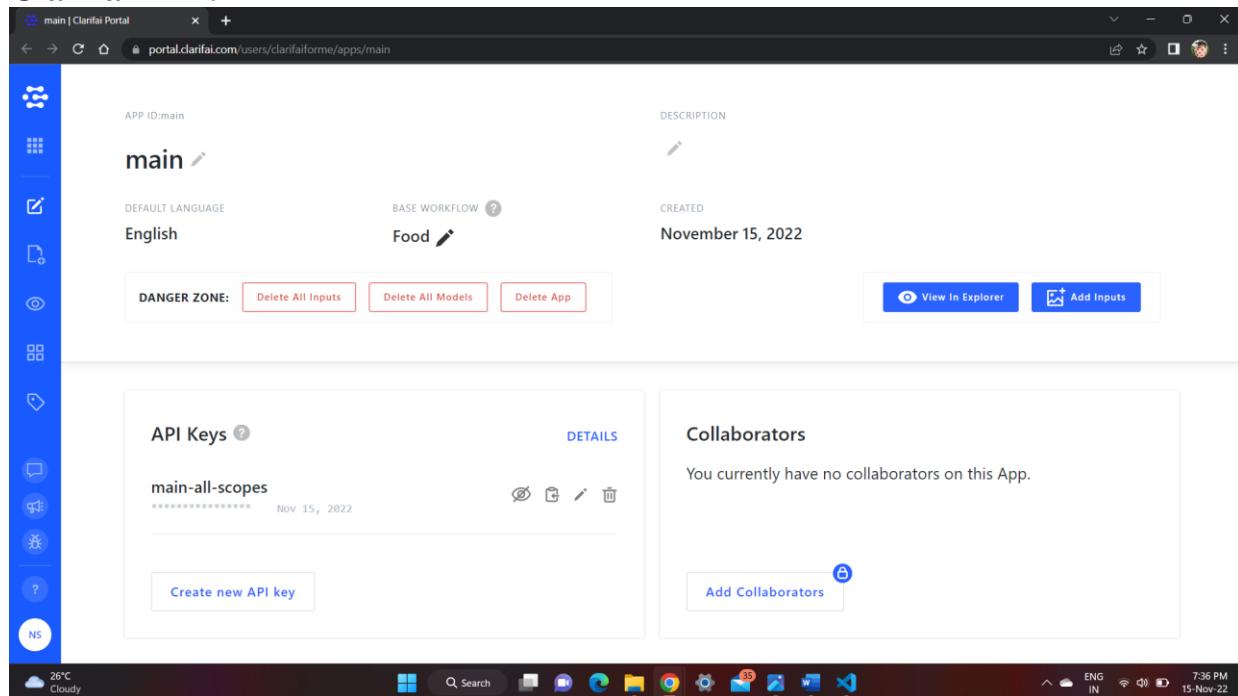
[Update Contact Information](#)

Profile

User-ID	Change User id
<input type="text" value="clarifaiforme"/>	

The bottom of the screen shows a Windows taskbar with the date and time as 8:23 PM on 15-Nov-22.

Clarifai API:



Python code to integrate with the Nutrition API:

```
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import resources_pb2, service_pb2, service_pb2_grpc
from clarifai_grpc.grpc.api.status import status_pb2, status_code_pb2
import os

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

metadata = (('authorization', 'Key ' + ' '),)

userDataObject = resources_pb2.UserAppIDSet(user_id='clarifaiforme',
app_id='main')

def food_identfier(string):
    with open(string, "rb") as f:
        file_bytes = f.read()

    post_model_outputs_response = stub.PostModelOutputs(
        service_pb2.PostModelOutputsRequest(
            user_app_id=userDataObject,
            model_id="food-item-recognition",
```

```

        version_id="1d5fd481e0cf4826aa72ec3ff049e044"
        inputs=[
            resources_pb2.Input(
                data=resources_pb2.Data(
                    image=resources_pb2.Image(
                        base64=file_bytes
                    )
                )
            )
        ],
        metadata=metadata
    )

    if post_model_outputs_response.status.code != status_code_pb2.SUCCESS:
        print("There was an error with your request!")
        print("\tCode:
{}".format(post_model_outputs_response.outputs[0].status.code))
        print("\tDescription:
{}".format(post_model_outputs_response.outputs[0].status.description))
        print("\tDetails:
{}".format(post_model_outputs_response.outputs[0].status.details))
        raise Exception("Post model outputs failed, status: " +
post_model_outputs_response.status.description)

    output = post_model_outputs_response.outputs[0]

    print("Predicted concepts:")
    for concept in output.data.concepts:
        print("\t%s %.2f" % (concept.name, concept.value))

    return output.data.concepts[0].name

```