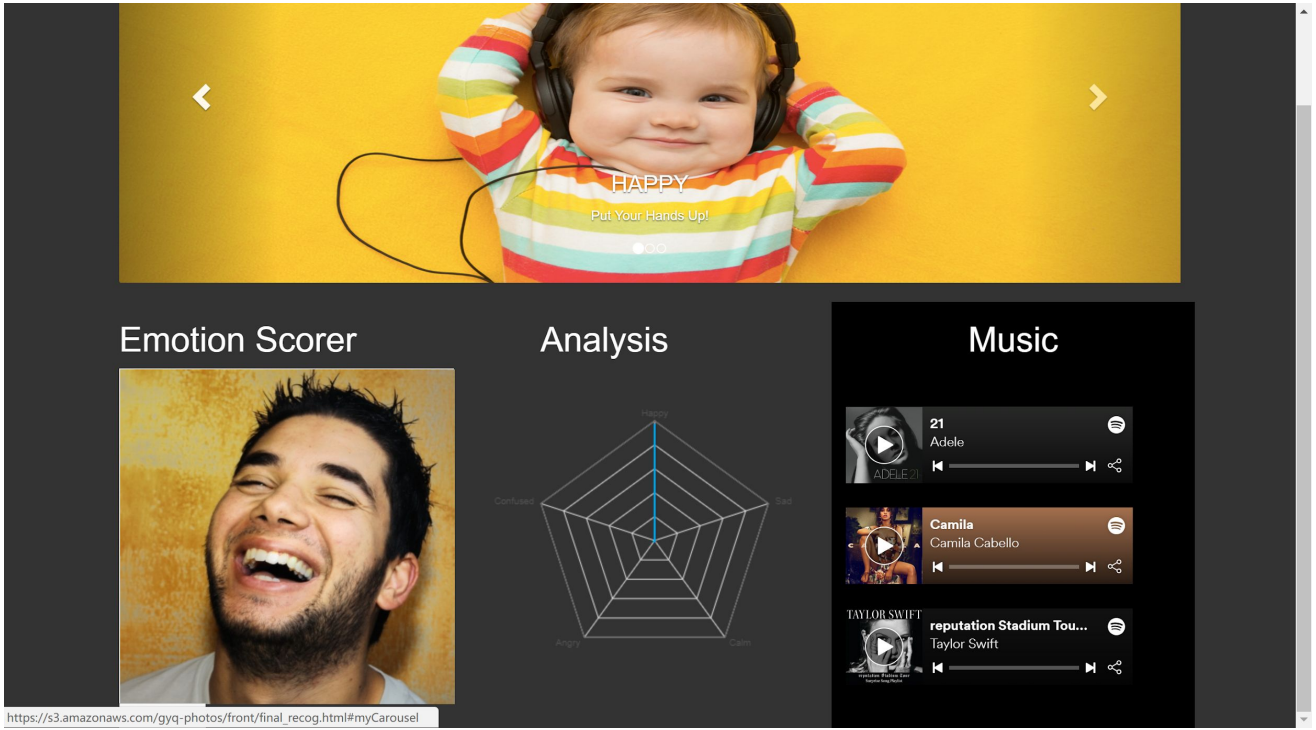
# MusicMood

yg2520 jw3535 pt2508 yh3052

# Our Purpose

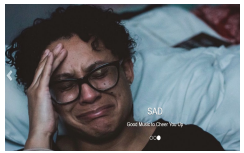
Want some music recommendation suit for your current emotion?


# Our Web Application



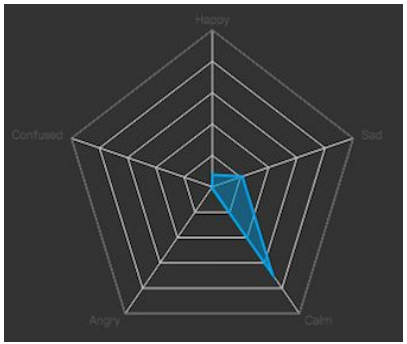
# Pipline

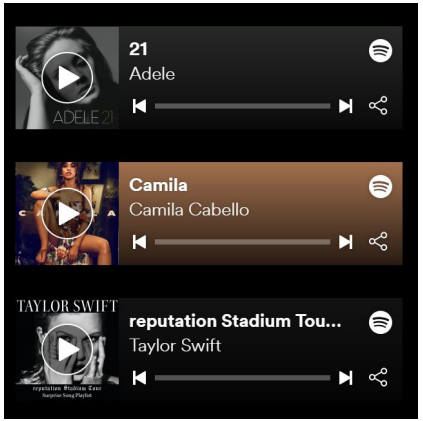
Facebook Login



Upload Your Photo

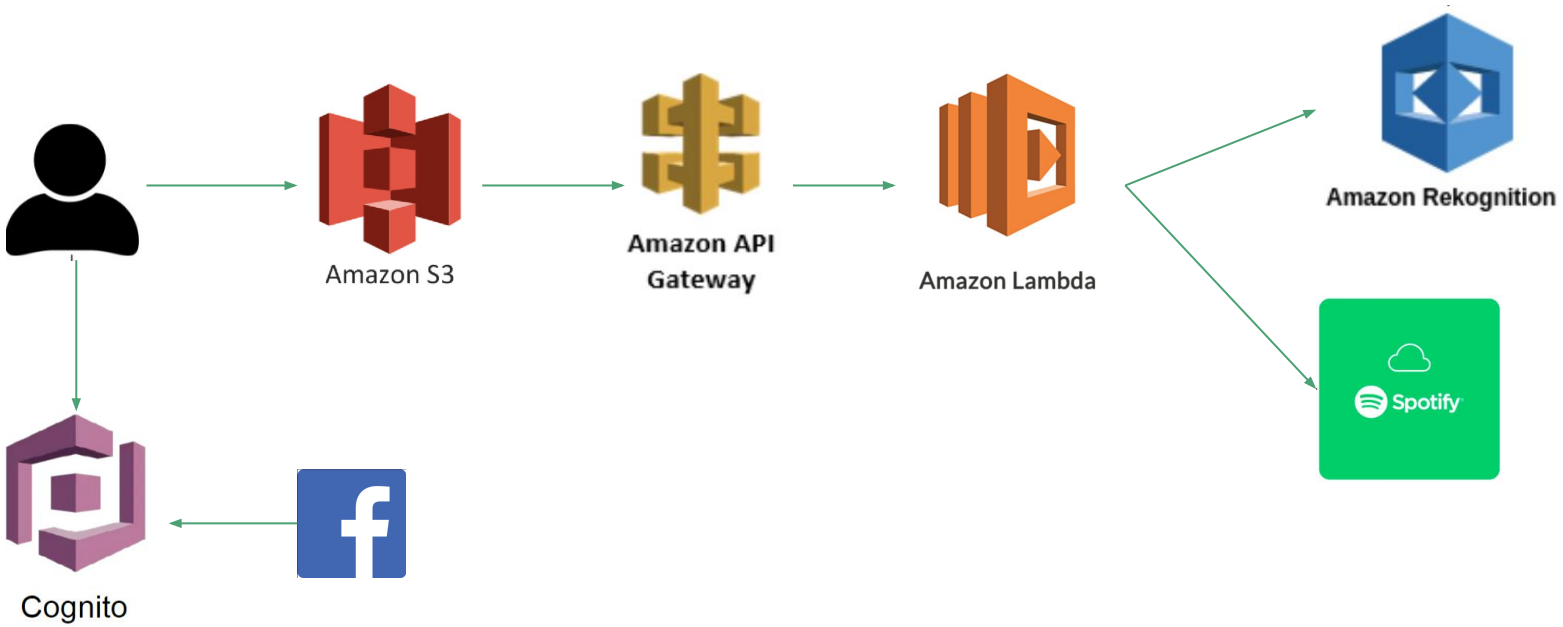
Get an emotion analysis



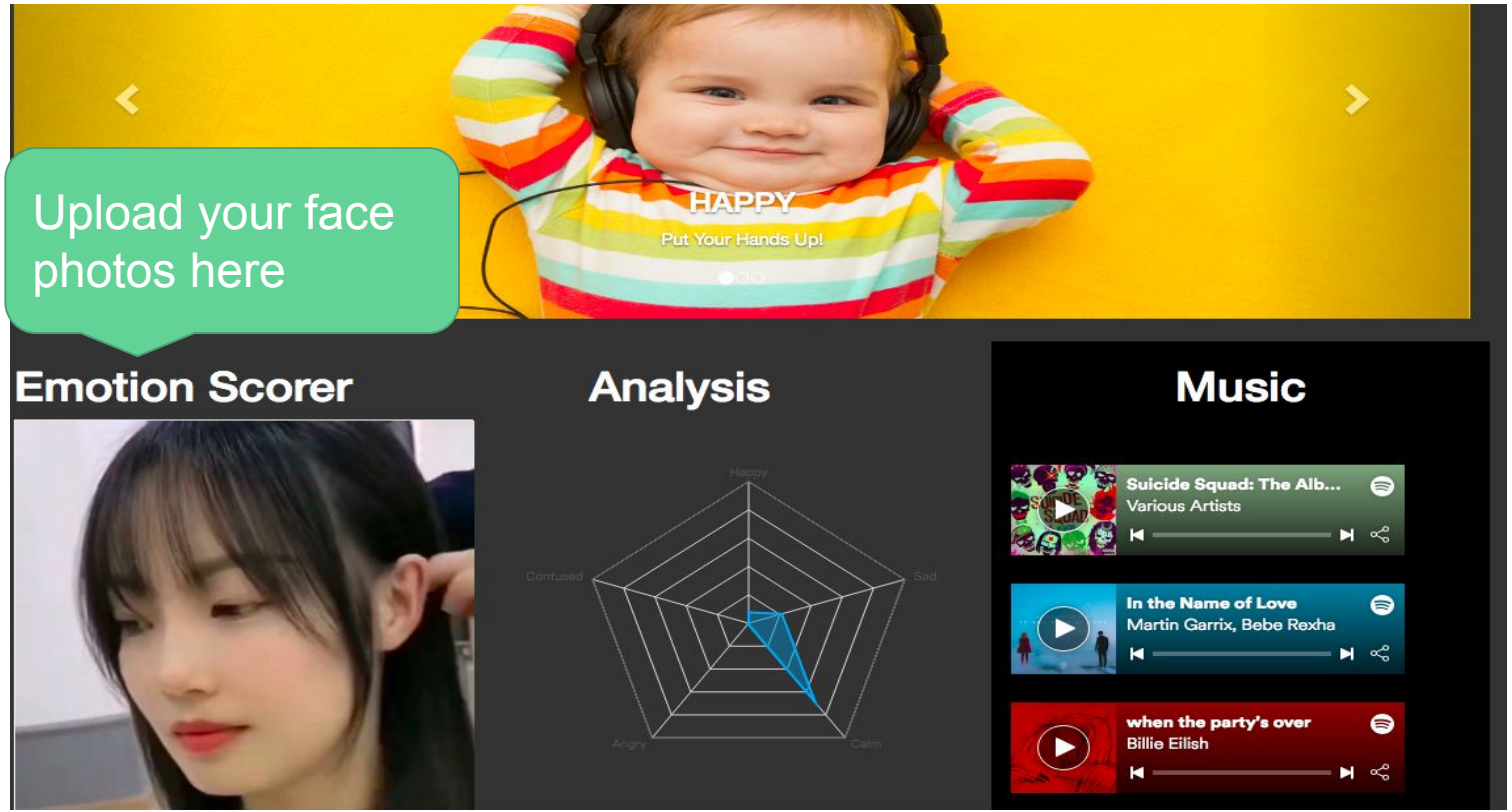




Our Music recommendation

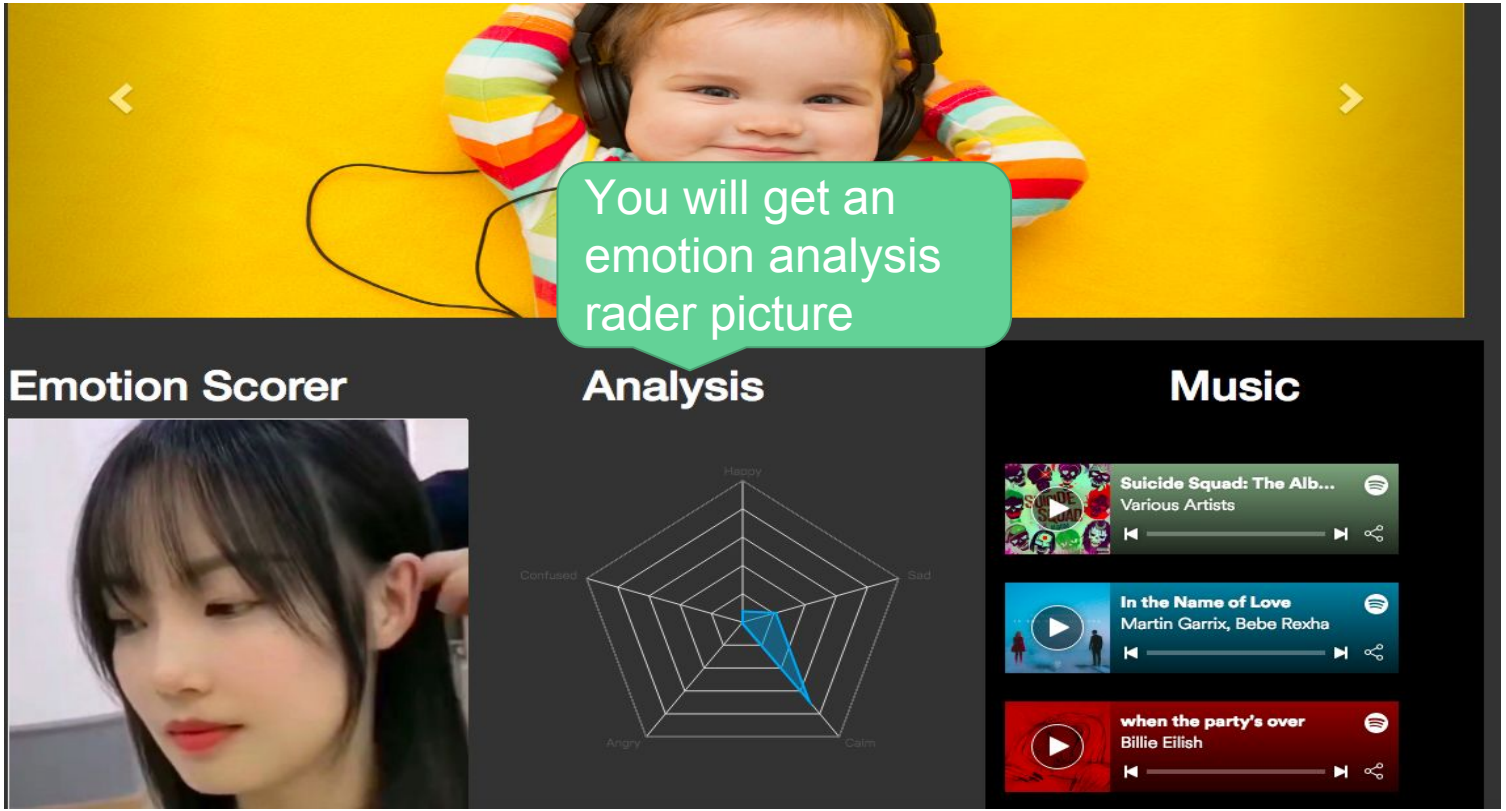
# Structure


Amazon S3
Amazon API Gateway
Amazon Lambda
Amazon Rekognition
Spotify
Cognito

# Frontend


HAPPY
Put Your Hands Up!
Upload your face photos here
Emotion Scorer
Analysis
Happy
Sad
Calm
Angry
Confused
Music
Suicide Squad: The Alb...
Various Artists
In the Name of Love
Martin Garrix, Bebe Rexha
when the party's over
Billie Eilish

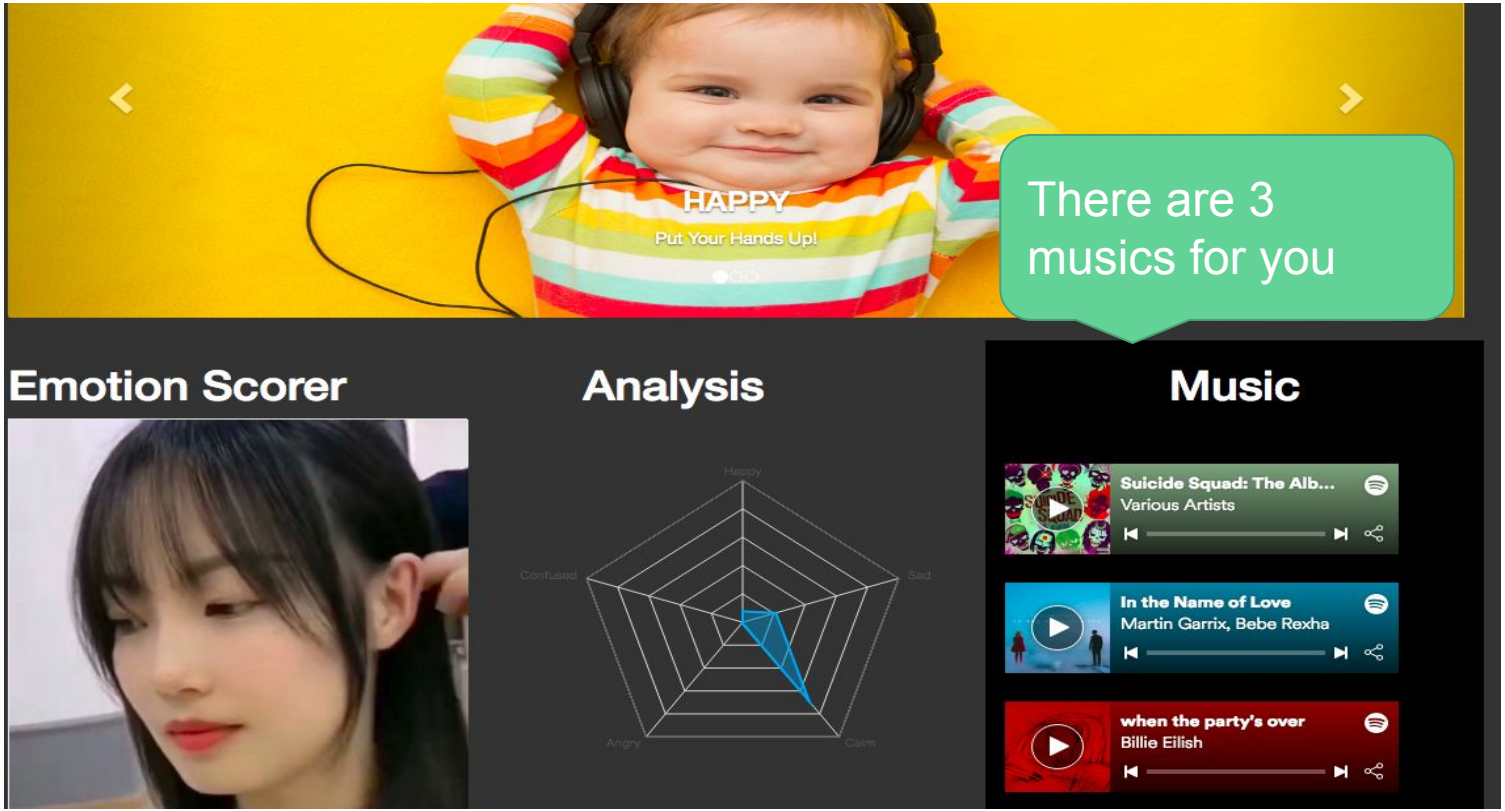
# Frontend


You will get an emotion analysis rader picture
Emotion Scorer
Analysis
Music
Happy
Sad
Calm
Angry
Confused
Suicide Squad: The Alb...
Various Artists
In the Name of Love
Martin Garrix, Bebe Rexha
when the party's over
Billie Eilish

# Frontend (code)

```
function redarLoad(happyScore, sadScore, clamScore, angryScore, confusedScore){
    var players = [
        { player: "User1",Happy: happyScore, Sad: sadScore, Calm: clamScore, Angry: angryScore, Confused: confusedScore}, ];
    var team = {Happy: 100, Sad: 100, Calm: 100, Angry: 100, Confused: 100 };
    var labels = [ [0, "Happy"], [1, "Sad"], [2, "Calm"], [3, "Angry"], [4, "Confused"] ];
    var get_player = function(name) {
           for (var i=0; i<players.length; i++) {
               if (players[i].player === name) return players[i];
           } }
    var player_data = function(name) {
        var obj = {}, i = 0;
        obj.label = name;
        obj.data = [];
        for (var key in team) {
            obj.data.push([i, get_player(name)[key]]);
            i++;
        };
        return obj;
    };
        Flotr.draw(
            document.getElementById("chart"),
            [
                player_data("User1")
            ],{
                title:"Emotion Analysis",
                radar:{show:true},
                legend:{show:false},
                grid:{circular:true,},
                xaxis:{ticks:labels, fontSize:15},
                yaxis:{showLabels:false,min:0,max:100},
```

# Frontend


HAPPY
Put Your Hands Up!
There are 3 musics for you
Emotion Scorer
Analysis
Happy
Sad
Calm
Angry
Confused
Music
Suicide Squad: The Alb...
Various Artists
In the Name of Love
Martin Garrix, Bebe Rexha
when the party's over
Billie Eilish

# Frontend (code)

```
function httpGet(happyScore)
{
    var xmlHttp = new XMLHttpRequest();
    xmlHttp.open( "GET", 'https://c1yoy0a9j5.execute-api.us-east-1.amazonaws.com/beta/search?happy='+happyScore, false);
    xmlHttp.send();
    console.log(xmlHttp.responseText);
    var musicJson = JSON.parse(xmlHttp.response);
    console.log(musicJson.tracks);
    //console.log(musicJson.tracks[0].album.uri.substring(0));
    var album1 = musicJson.tracks[0].album.uri.substring(14);
    document.getElementById("music1").innerHTML = '<iframe src="https://open.spotify.com/embed/album/'+album1+'" width="
    var album2 = musicJson.tracks[1].album.uri.substring(14);
    document.getElementById("music2").innerHTML = '<iframe src="https://open.spotify.com/embed/album/'+album2+'" width="
    var album3 = musicJson.tracks[2].album.uri.substring(14);
    document.getElementById("music3").innerHTML = '<iframe src="https://open.spotify.com/embed/album/'+album3+'" width="
```

# Backend:  Emotion analysis API

Amazon Rekognition

Detectface API

## Emotion Scorer

选择文件 sad1.jpg

| Happy | Sad | Calm | Angry | Confused |
|---|---|---|---|---|
| 4.83 | 4.66 | 71.18 | 4.33 | 0.00 |

# Backend:  Emotion analysis API

```
//Calls DetectFaces API and shows estimated ages of detected faces

function DetectFaces(imageData) {
  AWS.region = "us-east-1";
  var rekognition = new AWS.Rekognition();
  var params = {
    Image: {
      Bytes: imageData
    },
    Attributes: [
      'ALL',
    ]
  };
  rekognition.detectFaces(params, function (err, data) {
    if (err) console.log(err, err.stack); // an error occurred
    else {
     var table = "<table><tr><th>Happy</th><th>Sad</th><th>Calm</th><th>Angry</th><th>Confused</th></tr>";
      // show each face and build out estimated age table
      for (var i = 0; i < data.FaceDetails.length; i++) {
        for (var j=0; j<data.FaceDetails[i].Emotions.length; j++) {
          if (data.FaceDetails[i].Emotions[j].Type=="HAPPY"){
            var happyScore = data.FaceDetails[i].Emotions[j].Confidence.toFixed(2);
          }
          if (data.FaceDetails[i].Emotions[j].Type=="SAD"){
            var sadScore = data.FaceDetails[i].Emotions[j].Confidence.toFixed(2);
          }
          if (data.FaceDetails[i].Emotions[j].Type=="CALM"){
            var clamScore = data.FaceDetails[i].Emotions[j].Confidence.toFixed(2);
          }
```

# Backend: Music recommendation API

Spotify

## Get Recommendations Based on Seeds

Create a playlist-style listening experience based on seed artists, tracks and genres.

Recommendations are generated based on the available information for a given seed entity and matched against similar artists and tracks. If there is sufficient information about the provided seeds, a list of tracks will be returned together with pool size details.

```python
import json
from botocore.vendored import requests

def lambda_handler(event, context):
    # TODO implement
    h = event['queryStringParameters']['happy']
    happy = (float(h)+1)/2
    client_id = "175f21eafc104cefac621120d257ae3f"
    client_secret = "1660d911f7f64b09a12eb45465e3fade"
    grant_type = 'client_credentials'
    body_params = {'grant_type' : grant_type}
    url='https://accounts.spotify.com/api/token'
    response=requests.post(url, data=body_params, auth = (client_id, client_secret))
    res = response.content.decode()
    access_token = json.loads(res)["access_token"]
    print(access_token)
    headers = {"Authorization": "Bearer " + access_token}
    url = 'https://api.spotify.com/v1/recommendations?limit=5'
    params = {
        'seed_genres':'pop',
        'seed_tracks':'0c6xIDDpzE81m2q797ordA',
        'max_danceability':happy,
        'max_valence':happy,
        'max_energy':happy,
        'market':"US"

    }
    response = requests.get(url,params=params,headers=headers)
    return {
        'statusCode': 200,
        'headers':{
                "Access-Control-Allow-Origin" : "*"
                },
        'body': response.content.decode()
    }
```

# Demo

https://s3.amazonaws.com/gyq-photos/front/final_recog.html

https://s3.amazonaws.com/gyq-photos/front/final_recog_cognito.html