

Image based nutrition Analysis (Nutrisis)

C Ashvath Narayanan



Artificial Intelligence

IBM Build-a-thon

ashvathofficial@gmail.com

|  |  |  |
| --- | --- | --- |
| SNo. | Content | Page no. |
| 1 | INTRODUCTION | 2 |
|  | 1.1 Overview | 2 |
|  | 1.2 Purpose | 2 |
| 2 | LITERATURE SURVEY | 3 |
|  | 2.1 Existing problem | 3 |
|  | 2.2 Proposed solution | 3 |
| 3 | THEORITICAL ANALYSIS |  |
|  | 3.1 Block diagram |  |
|  | 3.2 Hardware / Software designing |  |
| 4 | EXPERIMENTAL INVESTIGATIONS |  |
| 5 | FLOWCHART |  |
| 6 | RESULT |  |
| 7 | ADVANTAGES & DISADVANTAGES |  |
| 8 | APPLICATIONS |  |
| 9 | CONCLUSION |  |
| 10 | FUTURE SCOPE |  |
| 11 | BIBILOGRAPHY |  |
| 12 | APPENDIX |  |
| 13 | SOURCE CODE |  |

Table of contents

1. Introduction

1.1 Overview

Improvement in people’s standards of living has led to obesity rates increasing at alarming speeds, and this is reflective to the risks in people’s health. People can avoid obesity by controlling their daily calorie intake and eating healthier foods. Although food packaging comes with nutrition (and calorie) labels, it’s still not very convenient for people to refer. Mobile-based nutrient dashboard systems which can analyses real time images of meal and analyze it for nutritional content can be very handy and improve the dietary habits, and therefore, result in healthy life.

1.2 Purpose

Nutrisis is primarily built for helping people monitor their food nutrition and calorie intakes. It should be able to recognize and analyze the food images inputted by the user. The application also has other features that interests people in maintaining a healthy life namely BMI (Body Mass Index) calculator, Daily Calorie calculator, etc.

2. Literature Survey

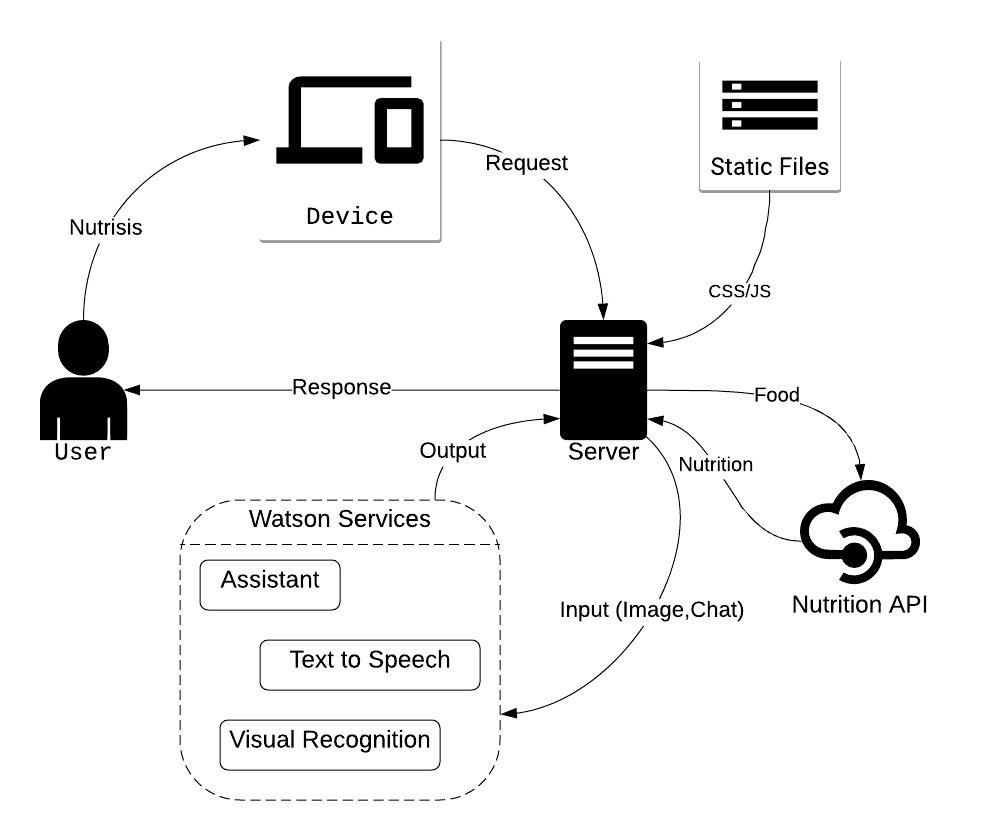
2.1 Existing Problem

People can calculate their calorie intake using the values mentioned on the packages of foods but not the food come in packages. There is no way to calculate calories for a dish ordered in a restaurant or a fully prepared meal on the table. Other nutrition apps can help do it but there are limits as different restaurant or manufacturers produce foods that vary in calories and ingredients. Knowing the calorie is important but different people have different calorie need depending on their BMI and workload. It is crucial to have knowledge of one’s daily calorie needs. People should also know their BMI and if their BMI is a healthy one or not. All these features have to be implemented in one application.

2.2 Proposed Solution

Nutrisis is an app fabricated for effective tracking of calorie and food intake for users. All users have to do is drop the image of their meal in the application and it will recognize the food and speak out its name and output all the various brands and types of the meal along with its respective nutritional and calorific values. The app has NutriBot, a chatbot which is used for calculating BMI, Daily calorie and talking about other food related information. This makes Nutrisis an interactive and user-friendly product.

3. Theoretical Analysis

3.1 Block diagram

3.2 Hardware/ Software designing

* Cloud object storage
* Visual recognition service
* Text to Speech service
* Watson Studio
* Watson Assistant
* IBM cloudant account
* Nutrition API (National Agriculture Library)
* Python
* Django web framework
* HTML
* CSS
* JS
* jQuery
* Bootstrap 4

4. Experimental Investigation

4.1 Creating IBM Cloud Account

* To create an IBM Cloud Account, go to <https://www.ibm.com/cloud> . Click on Create an account. If an account exists log in with the IBM id and password.

We create the following services after logging in:

* Visual Recognition

Search for Visual Recognition in the catalog. Select the service, choose a location and create an instance.

* Text to Speech

Search for Text to Speech in the catalog. Select the service, choose a location and create an instance.

* Watson Assistant

Search for Watson Assistant in the catalog. Select the service, choose a location and create an instance. Create an assistant (NutriBot) and add necessary skills (BMI, Daily Calorie Calculators)

* Cloud Object Storage

Search for Cloud Object Storage

in the catalog. Select the service, choose a location and create an instance.

Note: The location must be the same for all services

4.2 Integrating the Services

We will have to copy the API key and URL for the first three services mentioned above to access them in python using requests module.

Steps:

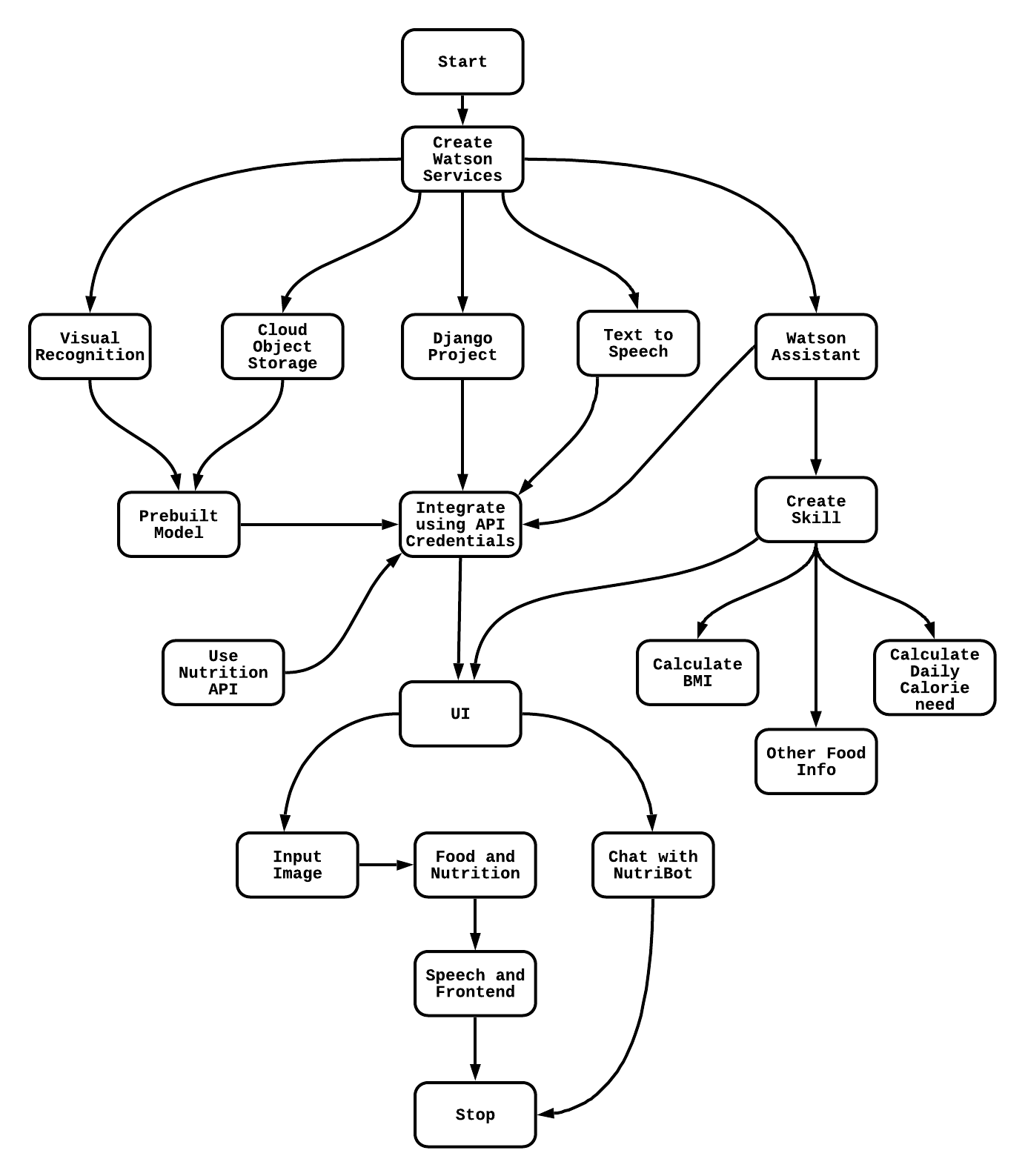
1. Create a Django project and design the UI for the application using templates.

* Create URL routes and respective functions
* Design using CSS and Bootstrap
* Add drag and drop image functionality using JS and jQuery
* Send data to frontend and show using template tags

1. Create the Watson Assistant with required skills.

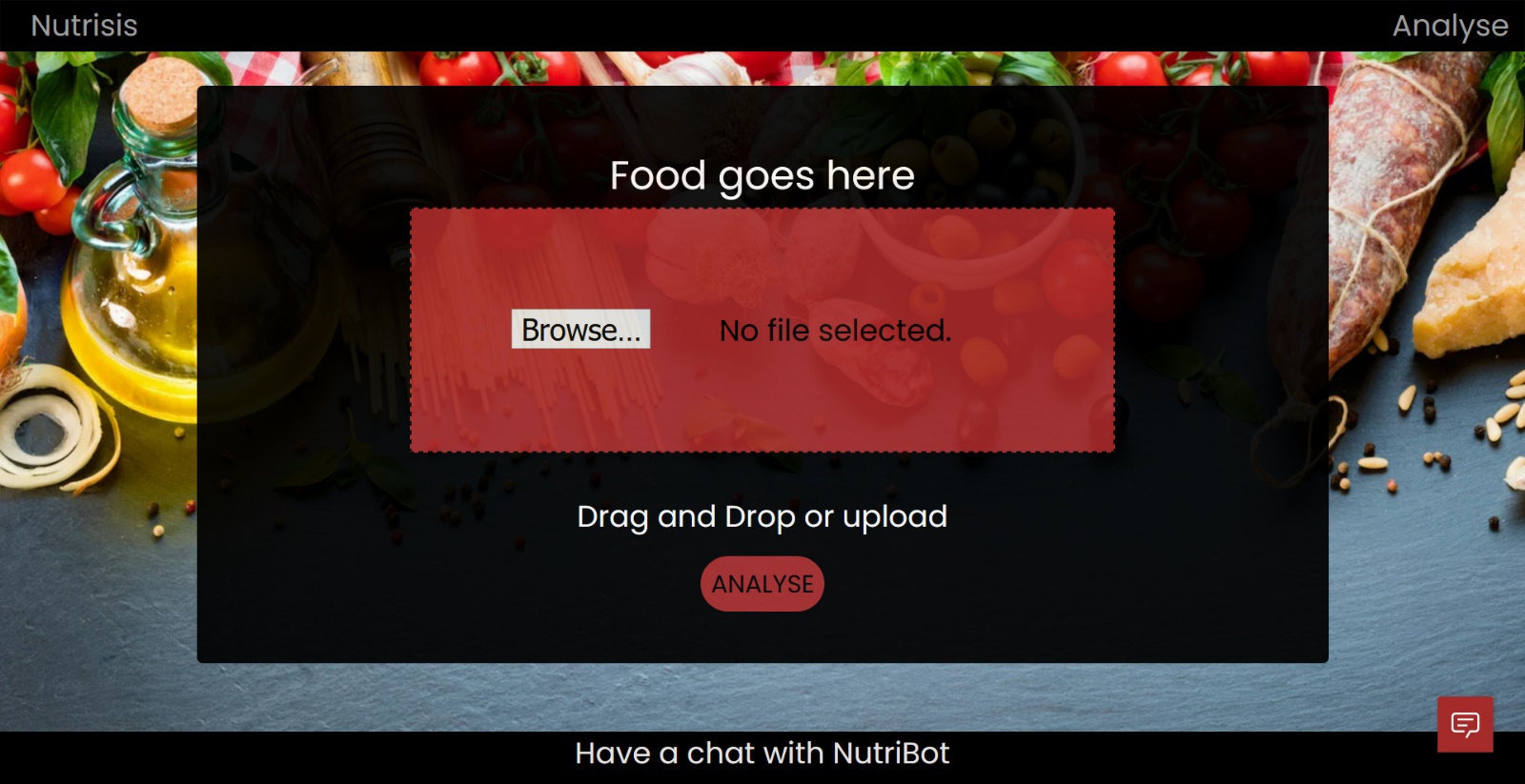
* Create intents, entities and dialogs
* Use system entities for height and weight
* Use slots for inputs
* Multiple conditioned responses for output
* Use child node to delete context variables
* Insert images using <img> tag

1. Use the prebuilt model for food recognition.
2. Connect with the Nutrition API (nal.gov.in)
3. Access the services in python using service credentials and requests module.

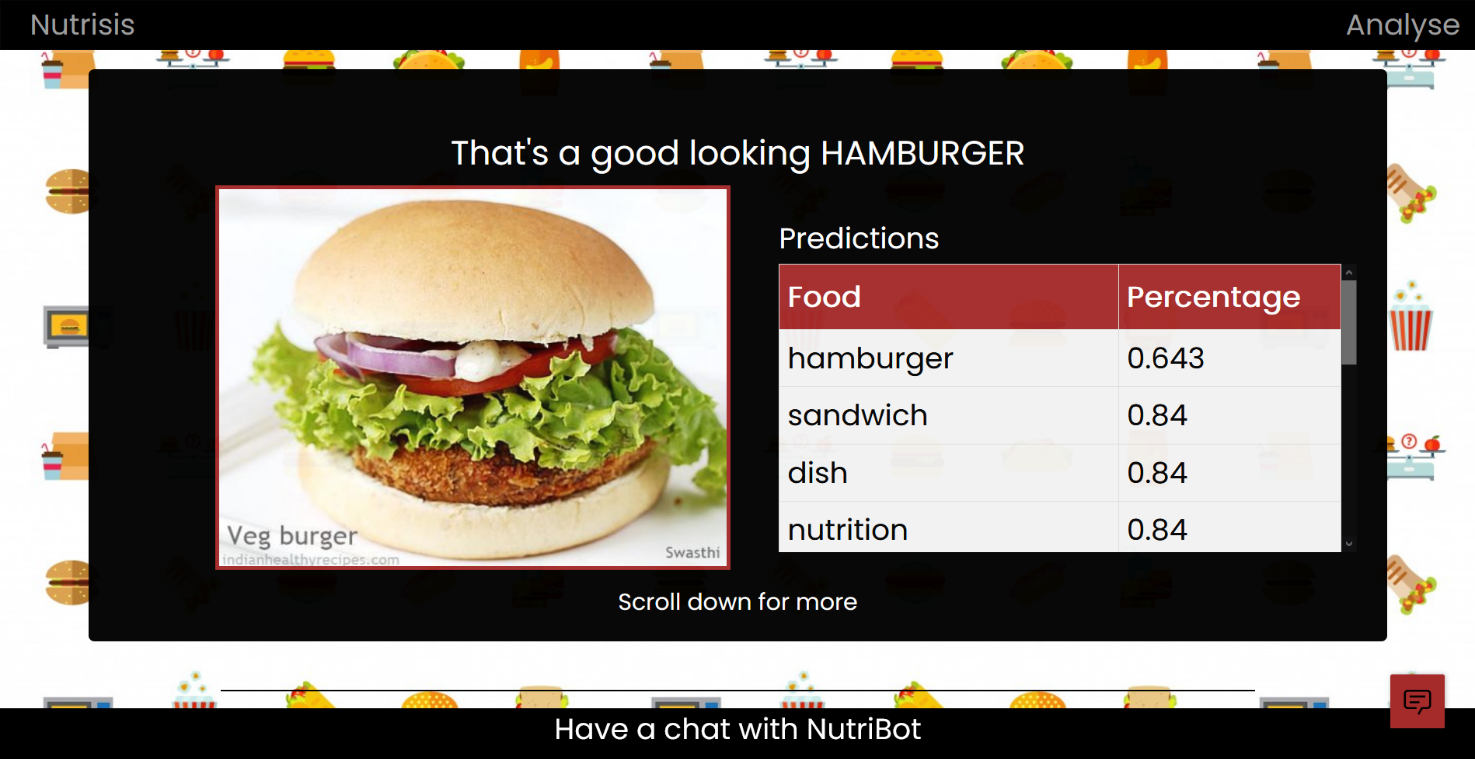
5. Flowchart

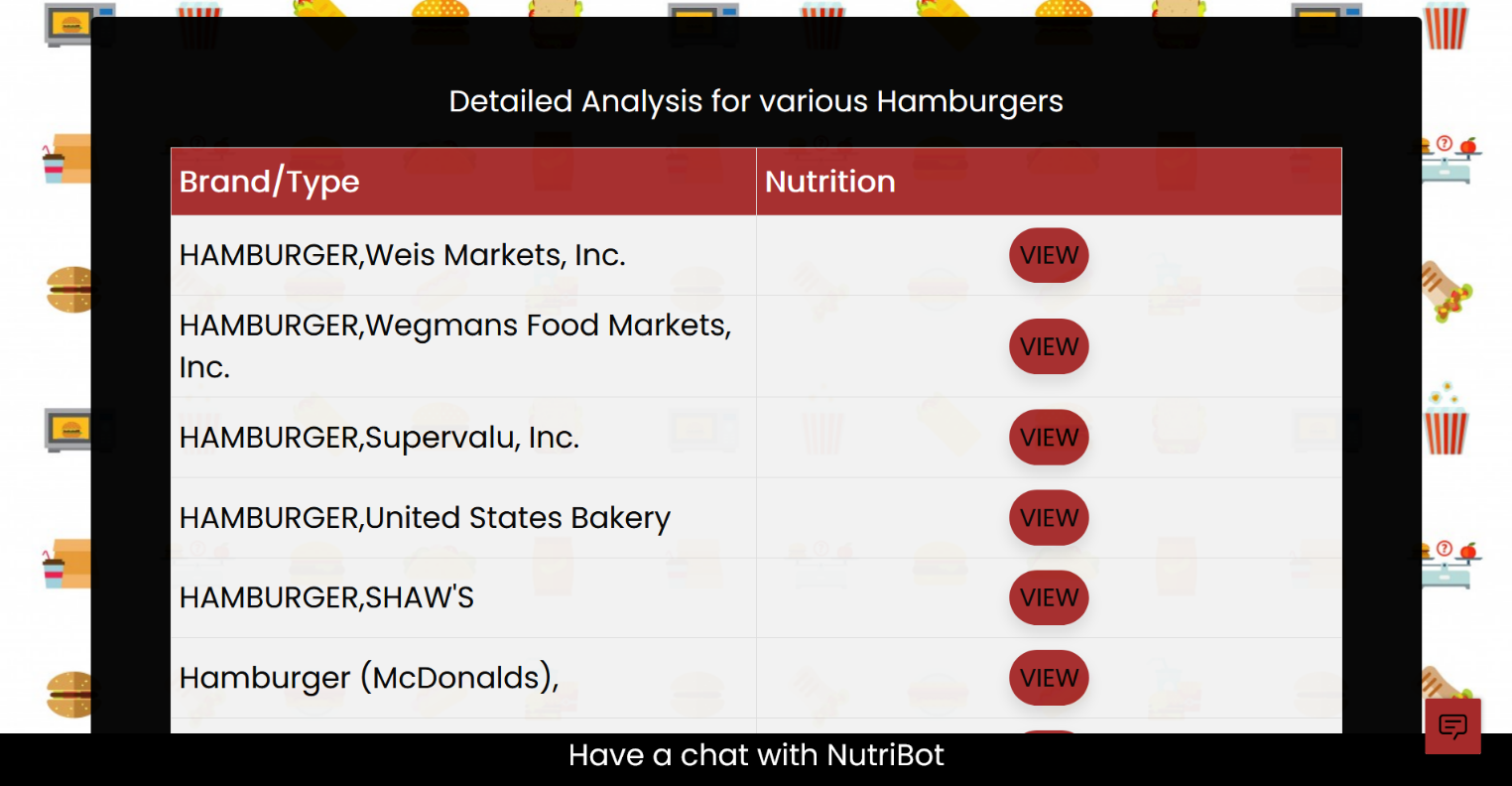
6. Results

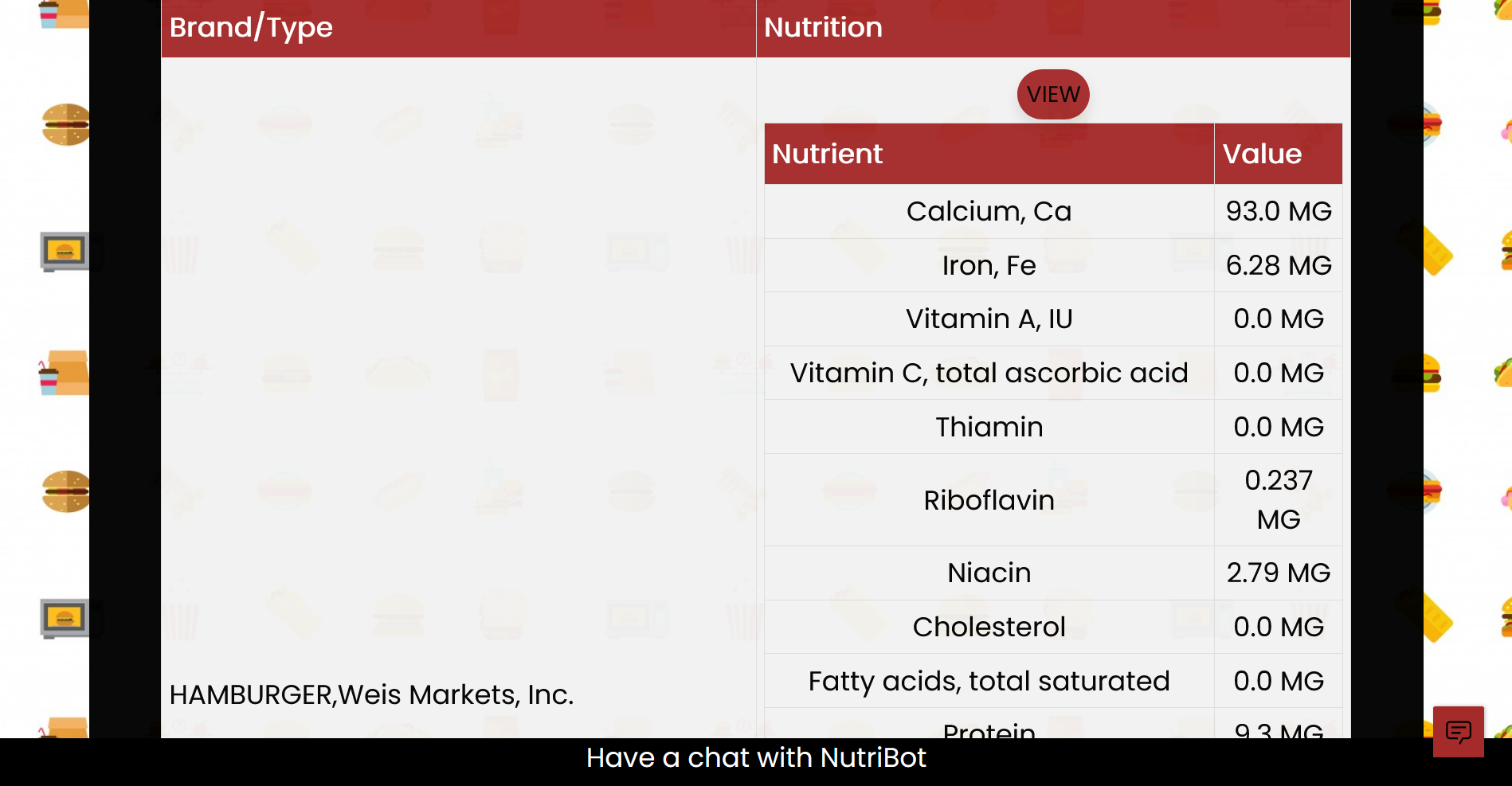
Landing Page:

Image input Page:

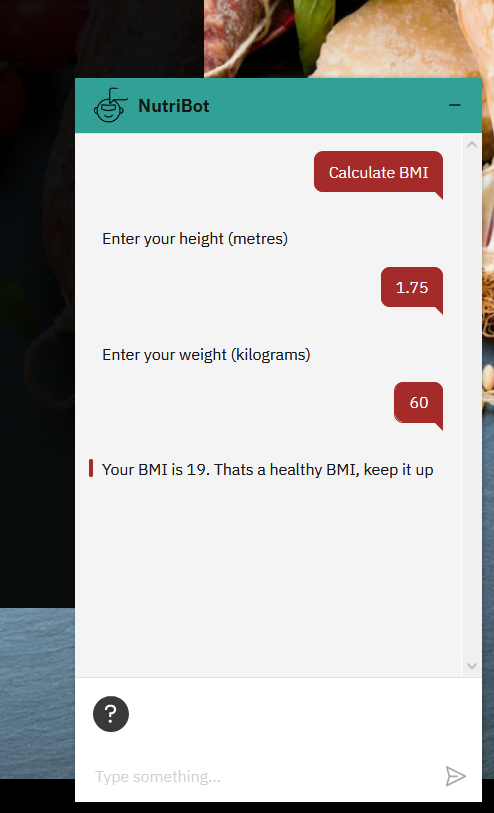
Result Page:

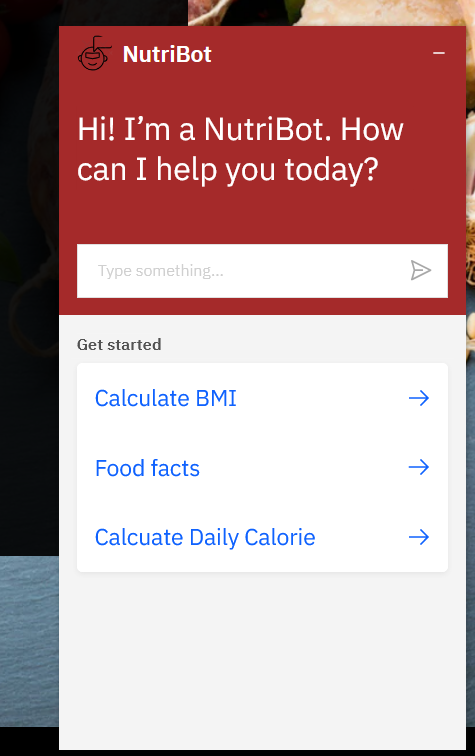






NutriBot:





7. Advantages and Disadvantages

Advantages

* Reusability
* User-friendly
* Available 24x7
* Makes calorie checking easy
* Can be accessed anywhere using device
* Secure
* Attractive UI/UX

Disadvantages

* Very slight chance of misprediction
* Depends on Internet

8. Applications

People can use this app for carrying out their diet plan wherever and whenever they want. Users can easily check their daily calorie needs and check whether their Body Mass Index is healthy or not. The app proves useful when people eat outside because the app gives nutrition values for various restaurant for a food.

9. Conclusions

Thus, by using Visual recognition, text to speech and Watson Assistant we have created a useful app for tracking and maintaining a good diet.

10.Future Scope

Future features include adding a schedule tracker and diet planner in the application. The accuracy of the model can be fine tuned and more data can be added.

11. Bibliography

<https://cloud.ibm.com/>

<https://smartinternz.com/>

<https://www.djangoproject.com/>

<https://stackoverflow.com/>

<https://github.com/>

<https://www.nal.usda.gov/main/>

Appendix: Source code

**Frontend:**

Land.html-

{%load static%}

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">

<title>Nutrisis</title>

<link rel="shortcut icon" href="{% static 'images/favicon2.png' %}">

</head>

<style media="screen">

body{

background-color: #010101;

background-image: url('https://www.freepik.com/blog/app/uploads/2018/06/food-cabacera-1-tasty-new.jpg');

}

\*{

font-family: 'Poppins', sans-serif;

}

.left {

padding-top: 220px;

padding-left: 120px;

animation: pulse 2s;

background-color:#e6e5e3;

opacity:0.9;

color: black;

height: 552px;

width: 1100px;

border-top-right-radius: 100%;

border-top-left-radius: 1%;

border-bottom-right-radius: 1%;

border-bottom-left-radius: 1%;

}

@keyframes pulse {

0% {

background-color:black;

border-top-right-radius: 0%;

border-top-left-radius: 0%;

border-bottom-right-radius: 0%;

border-bottom-left-radius: 0%;

padding-left: 0px;

opacity: 0;

}

100% {

background-color:#e6e5e3;

border-top-right-radius: 100%;

border-top-left-radius: 1%;

border-bottom-right-radius: 1%;

border-bottom-left-radius: 1%;

padding-left: 120px;

opacity: 0.9;

}

}

#h1{

font-size: 100px;

}

#h3{

font-size: 40px;

}

#h4{

border: none;

color: black;

text-decoration: none;

display: inline-block;

margin: 4px 2px;

cursor: pointer;

border-radius: 16px;

padding:10px;

font-size: 25px;

background-color: brown;

text-transform: uppercase;

font-weight: 500;

border: none;

border-radius: 45px;

box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.1);

transition: all 0.3s ease 0s;

cursor: pointer;

outline: none;

}

#h4:hover {

background-color: #2EE59D;

box-shadow: 0px 15px 20px rgba(46, 229, 157, 0.4);

color: #fff;

transform: translateY(-7px);

}

</style>

<body>

<div class="left">

<span id="h1"> Nutrisis </span><br>

<span id="h3"> Know your food better than ever </span><br><br>

<a href="{% url 'index' %}"><span id="h4"> Get Started</span><br></a>

</div>

</body>

</html>

Index.html

{%load static%}

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">

<title>Nutrisis</title>

<link rel="shortcut icon" href="{% static 'images/favicon2.png' %}">

</head>

<body>

<style media="screen">

\*{

font-family: 'Poppins', sans-serif;

font-size: 30px;

}

.footer {

position: fixed;

left: 0;

bottom: 0;

width: 100%;

background-color: black;

color: #e6e5e3;

text-align: center;

}

@keyframes pulse {

0% {

margin-top: -100px;

opacity: 0;

}

100% {

margin-top: 15px;

opacity: 0.92;

}

}

#image{

text-align: center;

padding: 100px;

border: 2px dashed black;

}

body{

background-image: url('https://www.azamara.com/sites/default/files/heros/med-food-hero.jpg')

}

.navbar{

background-color: black;

border-radius: 0px;

}

.jumbotron{

background-color: black;

color: white;

margin-top: 15px;

animation: pulse 2s;

opacity: 0.92;

}

#h4{

border: none;

color: black;

text-decoration: none;

display: inline-block;

margin: 4px 2px;

cursor: pointer;

border-radius: 16px;

padding:10px;

font-size: 25px;

background-color: brown;

text-transform: uppercase;

font-weight: 500;

border: none;

border-radius: 45px;

box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.1);

transition: all 0.3s ease 0s;

cursor: pointer;

outline: none;

}

#h4:hover {

background-color: #2EE59D;

box-shadow: 0px 15px 20px rgba(46, 229, 157, 0.4);

color: #fff;

transform: translateY(-7px);

}

input{

color:black;

background-color: brown;

}

h2{

font-size: 40px;

}

</style>

<nav class="navbar navbar-inverse">

<div class="container-fluid">

<ul class="nav navbar-nav navbar-left">

<li><a href="{%url 'land'%}">Nutrisis</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="{%url 'index'%}">Analyse</a></li>

</ul>

</div>

</nav>

<div class="container">

<div class="jumbotron">

<center>

<h2>Food goes here</h2>

<form method="post" enctype="multipart/form-data" action="{% url 'index'%}" method="post">

{%csrf\_token%}

{{form}} <p></p>

<input type="submit" id='h4' value="Analyse">

</form>

</center>

</div>

</div>

<div class="footer">

<p>Have a chat with NutriBot</p>

</div>

<script>

window.watsonAssistantChatOptions = {

integrationID: "47284a82-3253-4966-ade7-307e8ca9e178", // The ID of this integration.

region: "eu-gb", // The region your integration is hosted in.

serviceInstanceID: "ab104d1c-163e-44b6-be4b-10e03ec8bf9e", // The ID of your service instance.

onLoad: function(instance) { instance.render(); }

};

setTimeout(function(){

const t=document.createElement('script');

t.src="https://web-chat.global.assistant.watson.appdomain.cloud/loadWatsonAssistantChat.js";

document.head.appendChild(t);

});

</script>

</body>

</html>

Result.html

{%load static%}

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">

<title>Nutrisis</title>

<link rel="shortcut icon" href="{% static 'images/favicon2.png' %}">

<style media="screen">

\*{

font-family: 'Poppins', sans-serif;

font-size: 30px;

color:black;

}

#analysis {

font-family: Arial, Helvetica, sans-serif;

border-collapse: collapse;

width: 100%;

}

#analysis td, #analysis th {

border: 1px solid #ddd;

padding: 8px;

}

#analysis tr{background-color: #f2f2f2;}

#analysis tr:hover {background-color: #ddd;}

#analysis th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: brown;

color: white;

}

.scroll

{

width:120%;

height: 300px;

overflow-x: auto;

}

#h4{

border: none;

color: black;

text-decoration: none;

display: inline-block;

margin: 4px 2px;

cursor: pointer;

border-radius: 16px;

padding:10px;

font-size: 25px;

background-color: brown;

text-transform: uppercase;

font-weight: 500;

border: none;

border-radius: 45px;

box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.1);

transition: all 0.3s ease 0s;

cursor: pointer;

outline: none;

}

#h4:hover {

background-color: #2EE59D;

box-shadow: 0px 15px 20px rgba(46, 229, 157, 0.4);

color: #fff;

transform: translateY(-7px);

}

body{

background-image: url('https://image.freepik.com/free-vector/unhealthy-food-icon-set\_1262-4658.jpg')

}

#nutris{

display: none;

}

.container{

width:90%;

color:white;

}

.jumbotron{

background-color: black;

opacity: 0.97;

}

.footer {

position: fixed;

left: 0;

bottom: 0;

width: 100%;

background-color: black;

color: white;

text-align: center;

}

hr{

width: 70%;

border: 1px solid black;

}

.navbar{

background-color: black;

border-radius: 0px;

}

img{

border: 4px solid brown;

height: 400px;

width:400px;

}

</style>

</head>

<body>

<nav class="navbar navbar-inverse">

<div class="container-fluid">

<ul class="nav navbar-nav navbar-left">

<li><a href="{%url 'land'%}">Nutrisis</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="{%url 'index'%}">Analyse</a></li>

</ul>

</div>

</nav>

<audio autoplay>

<source src="{%static 'audio/now.wav'%}" type="audio/wav">

</audio>

<center>

<div class="container">

<div class="jumbotron">

<h1 style="font-size:35px;">That's a good looking {{result.0.class|upper}}</h1>

<p></p> <table>

<tr>

<td>

<img src={{image.image.url}} alt="" style="width:100%">

</td>

<td style="width:50px;">

</td>

<td>

<h2 style="color:white">Predictions</h2>

<div class="scroll">

<table id='analysis'>

<tr>

<th>Food</th>

<th>Percentage</th>

</tr>

{%for i in result%}

<tr>

<td>{{i.class}}</td>

<td>{{i.score}}</td>

</tr>

{%endfor%}

</table>

</div></td>

</tr>

</table>

<h3 style="color:white;margin-bottom:-20px">Scroll down for more</h3>

</div>

</div>

<hr>

<div class="container">

<div class="jumbotron">

<h2>Detailed Analysis for various {{result.0.class|title}}s</h2>

<div style="padding:20px">

<table id="analysis">

<th id='head'>Brand/Type</th>

<th>Nutrition</th>

{%for i in nutrition%}

{%if i.fdcId%}

<tr>

<td style="width:50%">{{i.description}},{{i.brandOwner}}</td>

<td style="text-align:center"><button id="h4" onclick="myFunction(this)">View</button>

<br>

<table id='nutris'>

<th>Nutrient</th>

<th>Value</th>

{%for j in i.foodNutrients%}

<tr>

<td>{{j.nutrientName}}</td>

<td>{{j.value}} MG</td>

</tr>

{%endfor%}

</table>

</td>

</tr>

{%endif%}

{%endfor%}

</table>

</div>

</div>

</div>

</center>

<div class="footer">

<p style="color:white">Have a chat with NutriBot</p>

</div>

<script type="text/javascript">

window.watsonAssistantChatOptions = {

integrationID: "47284a82-3253-4966-ade7-307e8ca9e178", // The ID of this integration.

region: "eu-gb", // The region your integration is hosted in.

serviceInstanceID: "ab104d1c-163e-44b6-be4b-10e03ec8bf9e", // The ID of your service instance.

onLoad: function(instance) { instance.render(); }

};

setTimeout(function(){

const t=document.createElement('script');

t.src="https://web-chat.global.assistant.watson.appdomain.cloud/loadWatsonAssistantChat.js";

document.head.appendChild(t);

});

function myFunction(obj) {

var x =obj.parentElement;

if (x.children[2].style.display == "block") {

x.children[2].style.display = "none";

} else {

x.children[2].style.display = "block";

}

}

</script>

</body>

</html>

**Backend**:

Python:

Views.py

from django.shortcuts import render, redirect

from django.http import HttpResponse

from .forms import \*

import json,requests,os

from watson\_developer\_cloud import VisualRecognitionV3

from ibm\_watson import TextToSpeechV1

from ibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

def land(request):

return render(request, 'land.html')

def index(request):

if request.method == 'POST':

form = ImageForm(request.POST, request.FILES)

if form.is\_valid():

obj=form.save()

return redirect('result',obj.id)

else:

form = ImageForm()

return render(request, 'index.html', {'form' : form})

def result(request,id):

authenticator = IAMAuthenticator('0gtx\_JpZ7asvTN5qh8fQBsaWefHqOE15Z0DZvN8BrZKV')

text\_to\_speech = TextToSpeechV1(authenticator=authenticator)

visual\_recognition = VisualRecognitionV3(

'2018-03-19',

iam\_apikey='tUr8iCtLPxhe-ZxzCc1lMttk0MiVfpBudbwQOzQtBMOR')

obj=Image.objects.get(id=id)

print(obj.image)

with open('media/'+str(obj.image), 'rb') as images\_file:

classes = visual\_recognition.classify(

images\_file,

threshold='0',

classifier\_ids='default').get\_result()

nutrition = requests.get('https://api.nal.usda.gov/fdc/v1/foods/search?api\_key=CcOcS7GaR1IkCaPsbFRgy95DBKw77LIGzZTyoIDd&query='+classes['images'][0]['classifiers'][0]['classes'][0]['class']).json()

text\_to\_speech.set\_service\_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/13febd73-d584-44ee-9407-f08f768e7773')

os.remove("static/audio/now.wav")

with open('static/audio/now.wav', 'wb') as audio\_file:

audio\_file.write(text\_to\_speech.synthesize("That's a good looking "+classes['images'][0]['classifiers'][0]['classes'][0]['class'],voice='en-US\_AllisonV3Voice',accept='audio/wav').get\_result().content)

return render(request, 'result.html',{'image':obj,'nutrition':nutrition["foods"],'result':classes['images'][0]['classifiers'][0]['classes']})

forms.py

from django import forms

from .models import \*

class ImageForm(forms.ModelForm):

image = forms.ImageField(widget=forms.FileInput(attrs={'id':'image'}))

class Meta:

model = Image

fields = ['image']

def \_\_init\_\_(self,\*args,\*\*kwargs):

super().\_\_init\_\_(\*args,\*\*kwargs)

self.fields['image'].label=''

self.fields['image'].help\_text='Drag and Drop or upload'

Urls.py

from django.contrib import admin

from django.urls import path

from app import views

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

path('admin/', admin.site.urls),

path('',views.land,name='land'),

path('feed',views.index,name='index'),

path('result/<id>',views.result,name='result')

]

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL,

document\_root=settings.MEDIA\_ROOT)