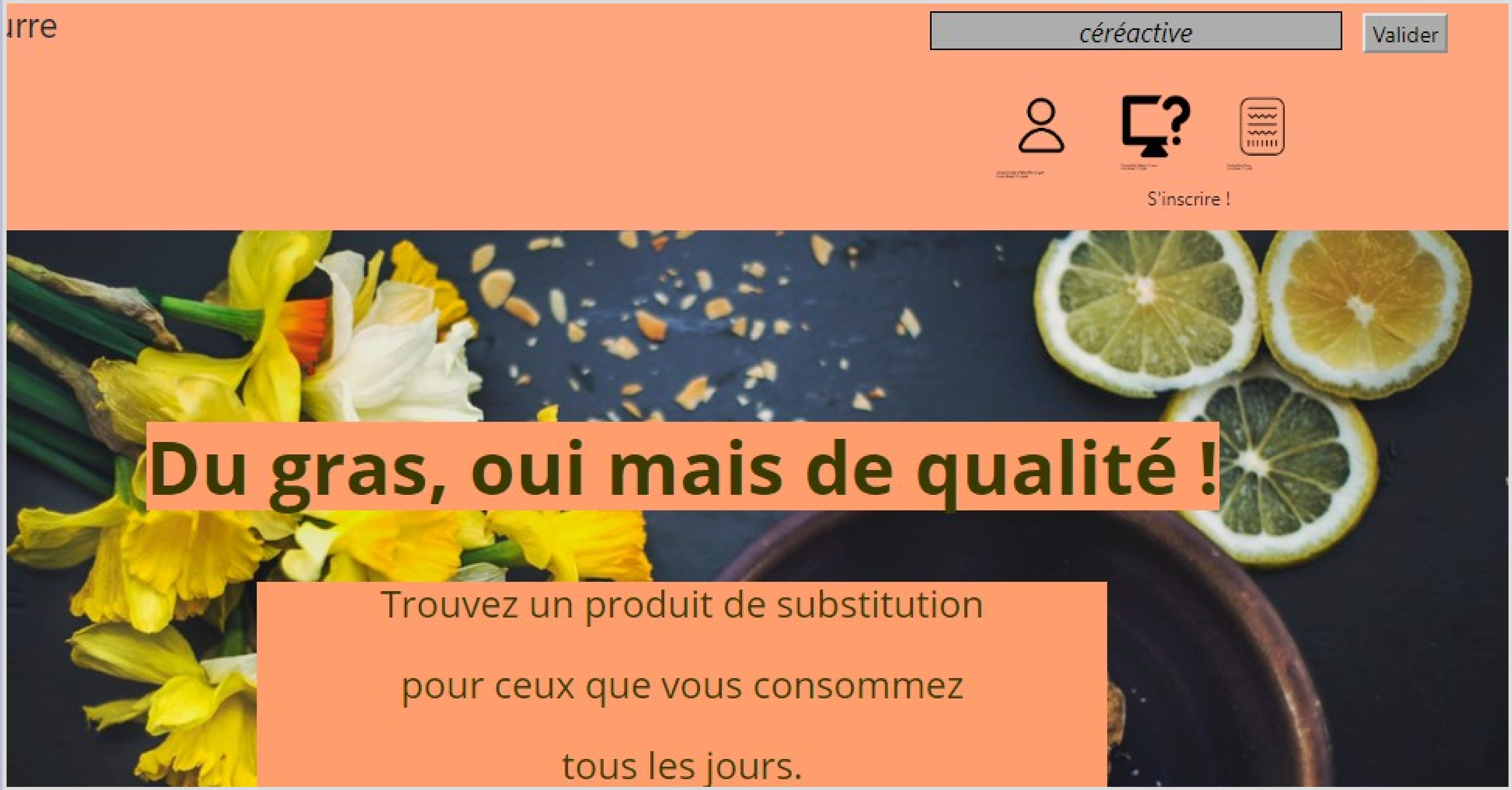
Projet 8

Plateforme nutella

Plateforme Nutella



Du gras, oui mais de qualité !

Trouvez un produit de substitution

pour ceux que vous consommez

tous les jours.

Plan

Plan

I Contexte

- Le site web

  - trouver des substituts plus sains à travers un site web.

- Plusieurs applications

  - connexion/inscription
  - recherche d'aliment
  - enregistrement
  - remplacement
  - visualisation

Plan

II Technologies utilisées

- différents langages de programmation

- HTML/CSS *(affichage visuel)*

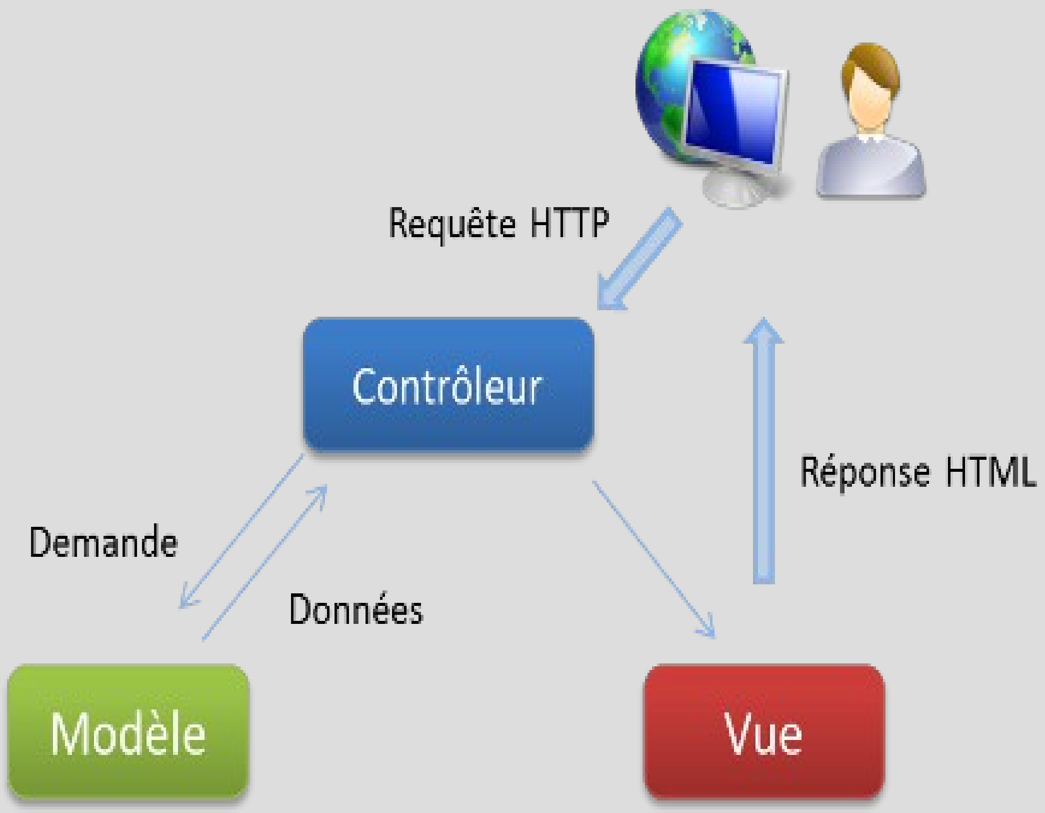
- Javascript/Jquerry *(animation front et requête asyncrhone)*

- Python *(Traitement de donnée)*

- framework Django *(Traitement de donnée)*

Plan

III Schéma de développement de Django



Plan

III Schéma de développement de Django

MODELE

- L'orm un outils pratique

  - gestion de la base de donnée

Plan

III Schéma de développement de Django

CONTROLEUR

- Les views

  - interaction ente les modèles et les templates

Plan

I contexte

II Technologies utilisées

III Schéma de développement de Django

- *Modèle*
- *Vues*
- *Templates*

IV Cheminement complet d'une requête

V Intégration continue (HEROKU)

VI Gestion de projet

III Schéma de développement de Django

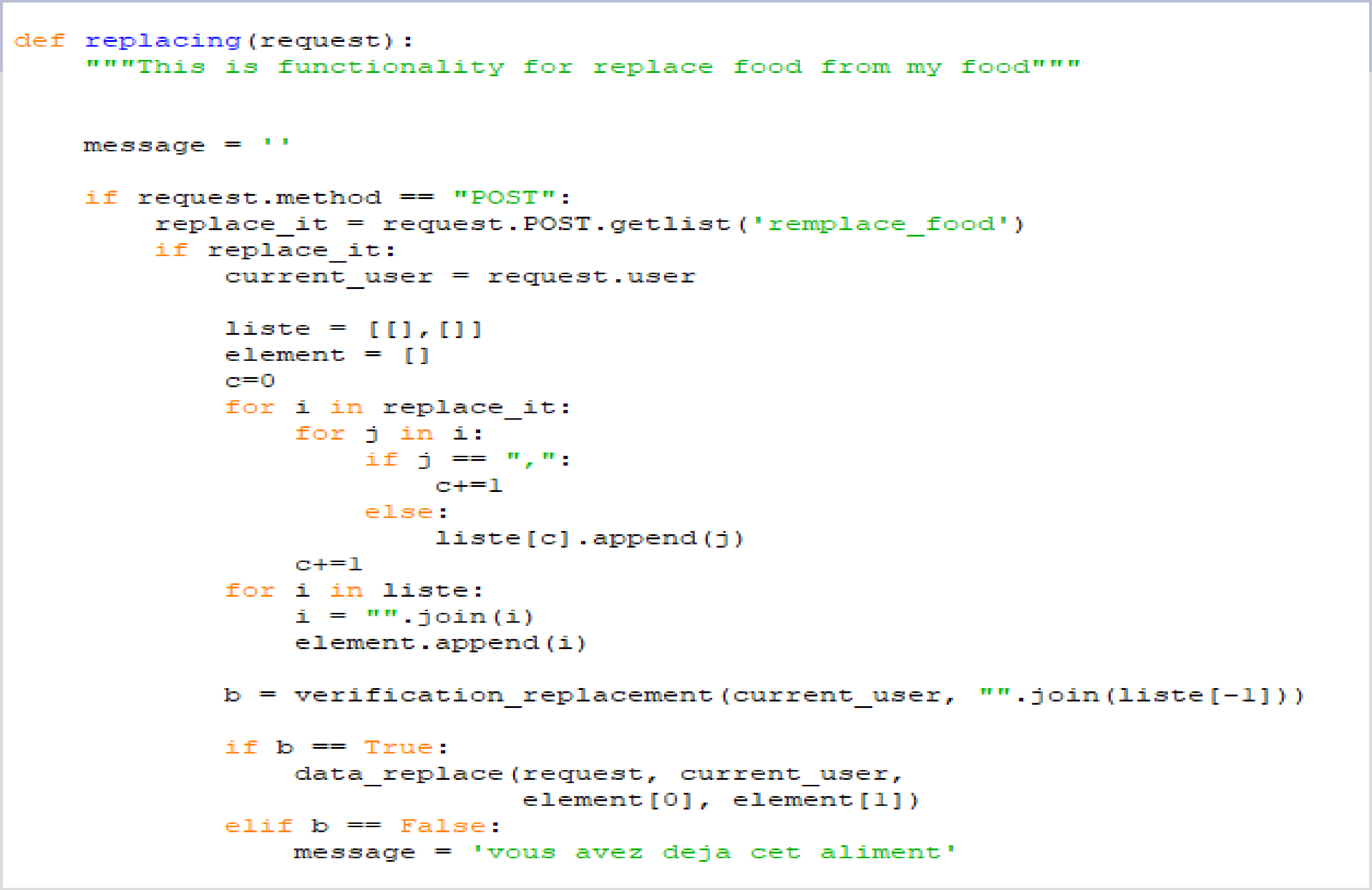
TEMPLATES

- Les templates

  - *restitution graphique générées par les vues*

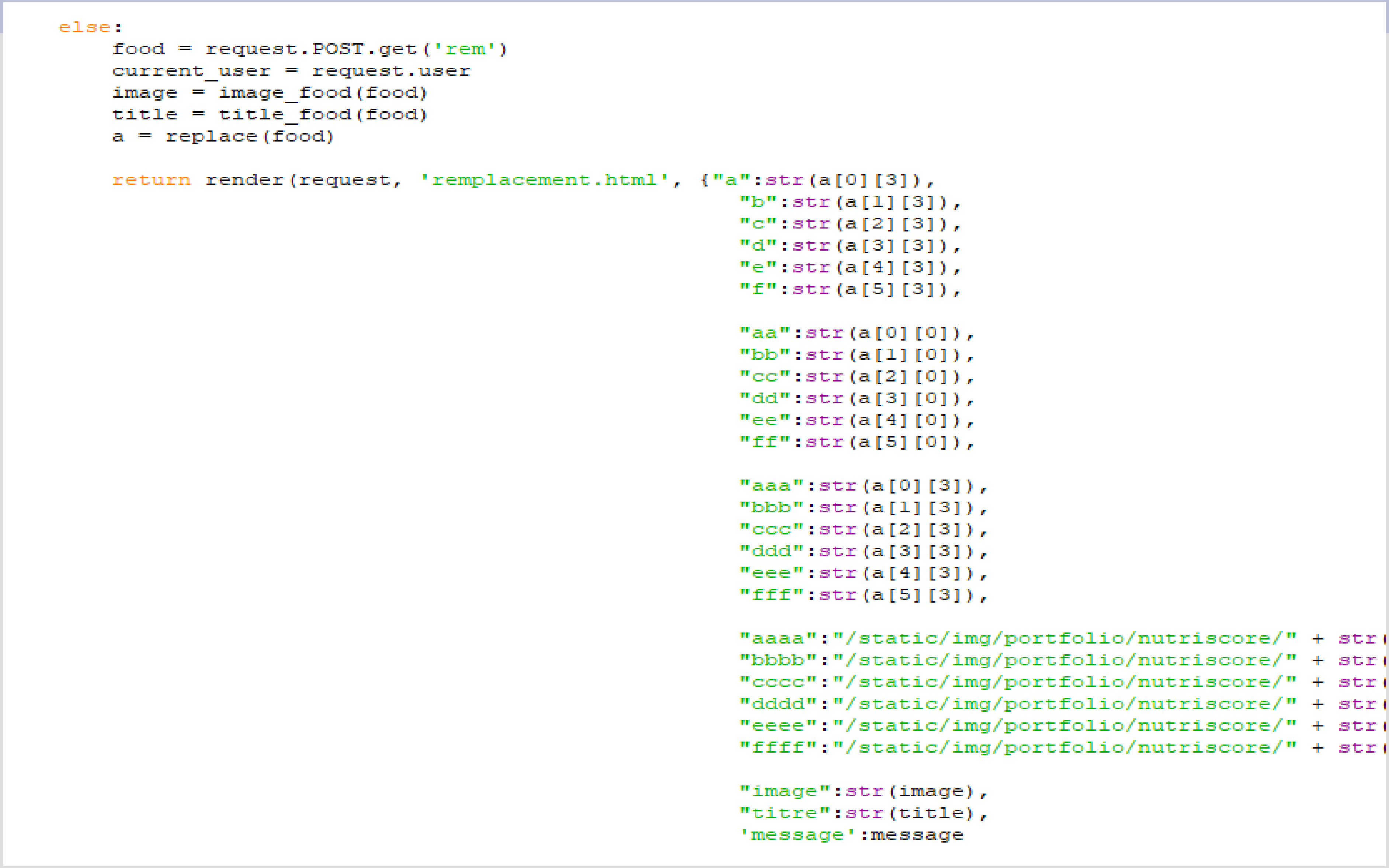
Plan

Vues - la fonction replacing()



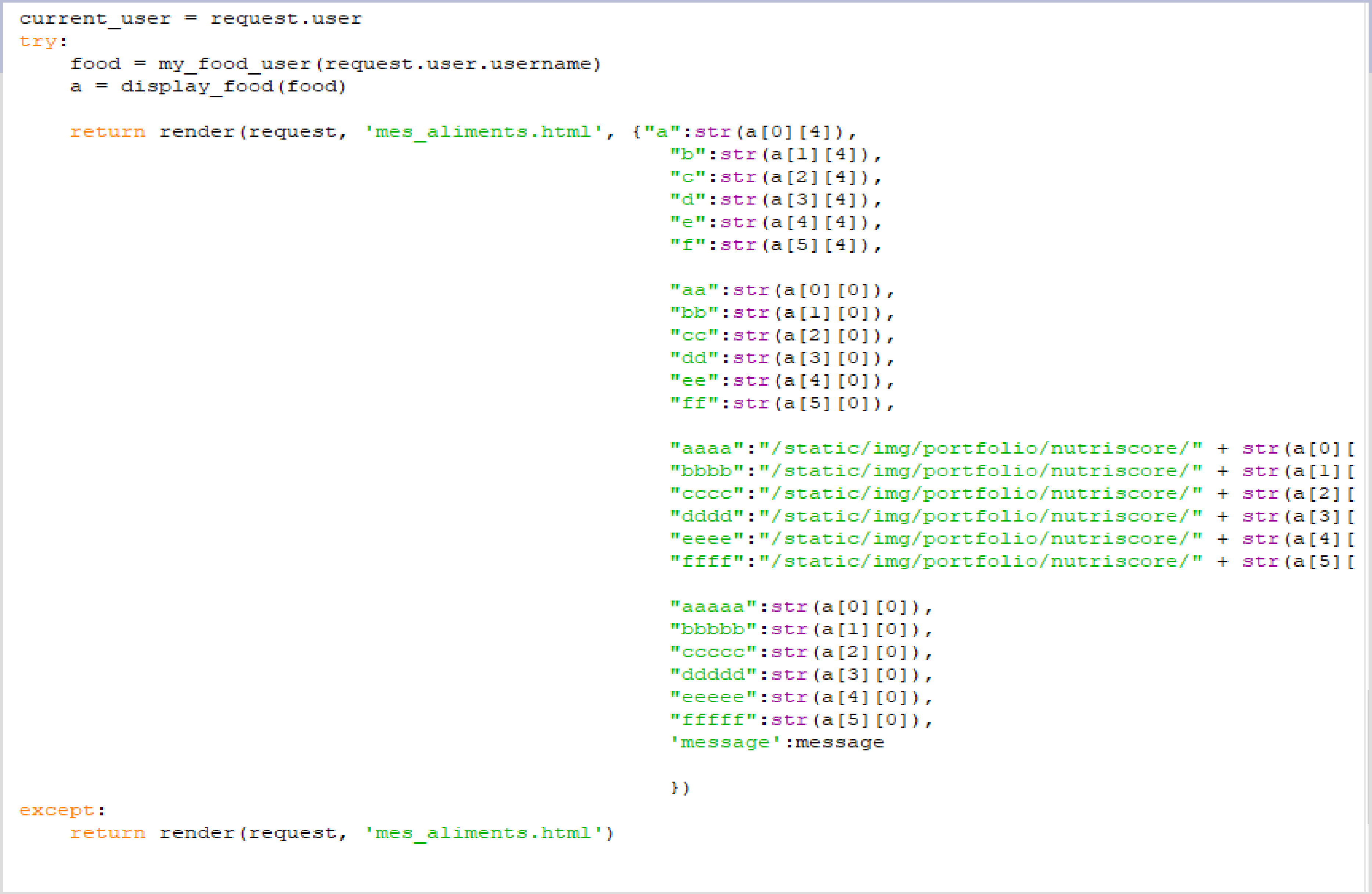
```python
def replacing(request):
    """This is functionality for replace food from my food"""


    message = ''

    if request.method == "POST":
        replace_it = request.POST.getlist('remplace_food')
        if replace_it:
            current_user = request.user

            liste = [[],[]]
            element = []
            c=0
            for i in replace_it:
                for j in i:
                    if j == ",":
                        c+=1
                    else:
                        liste[c].append(j)
                c+=1
            for i in liste:
                i = "".join(i)
                element.append(i)

            b = verification_replacement(current_user, "".join(liste[-1]))

            if b == True:
                data_replace(request, current_user,
                             element[0], element[1])
            elif b == False:
                message = 'vous avez deja cet aliment'
```

Vues - la fonction replacing



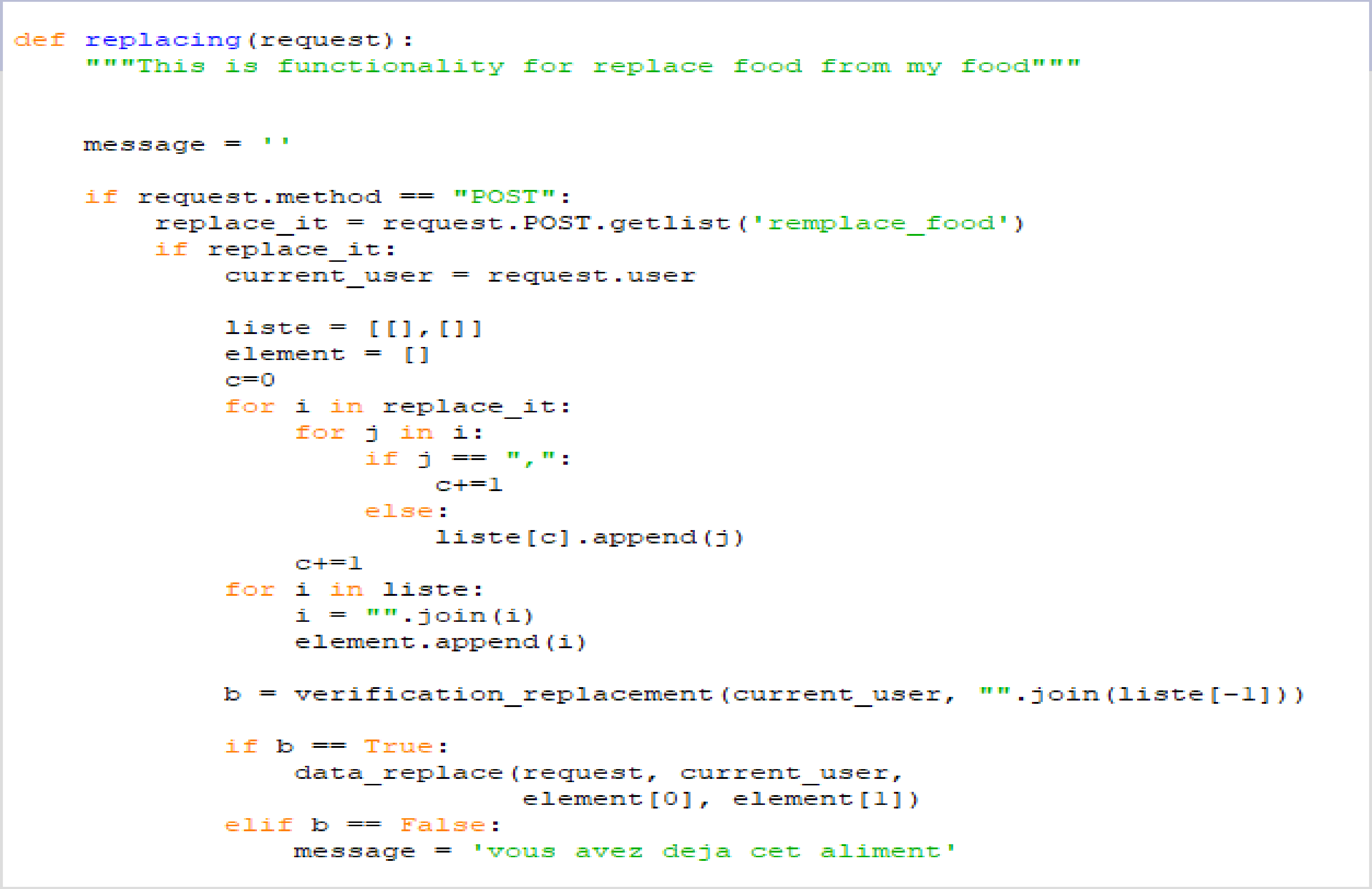
```
else:
    food = request.POST.get('rem')
    current_user = request.user
    image = image_food(food)
    title = title_food(food)
    a = replace(food)

    return render(request, 'remplacement.html', {"a":str(a[0][3]),
                                                   "b":str(a[1][3]),
                                                   "c":str(a[2][3]),
                                                   "d":str(a[3][3]),
                                                   "e":str(a[4][3]),
                                                   "f":str(a[5][3]),

                                                   "aa":str(a[0][0]),
                                                   "bb":str(a[1][0]),
                                                   "cc":str(a[2][0]),
                                                   "dd":str(a[3][0]),
                                                   "ee":str(a[4][0]),
                                                   "ff":str(a[5][0]),

                                                   "aaa":str(a[0][3]),
                                                   "bbb":str(a[1][3]),
                                                   "ccc":str(a[2][3]),
                                                   "ddd":str(a[3][3]),
                                                   "eee":str(a[4][3]),
                                                   "fff":str(a[5][3]),

                                                   "aaaa":"/static/img/portfolio/nutriscore/" + str(
                                                   "bbbb":"/static/img/portfolio/nutriscore/" + str(
                                                   "cccc":"/static/img/portfolio/nutriscore/" + str(
                                                   "dddd":"/static/img/portfolio/nutriscore/" + str(
                                                   "eeee":"/static/img/portfolio/nutriscore/" + str(
                                                   "ffff":"/static/img/portfolio/nutriscore/" + str(

                                                   "image":str(image),
                                                   "titre":str(title),
                                                   'message':message
```

Vues - la fonction replacing



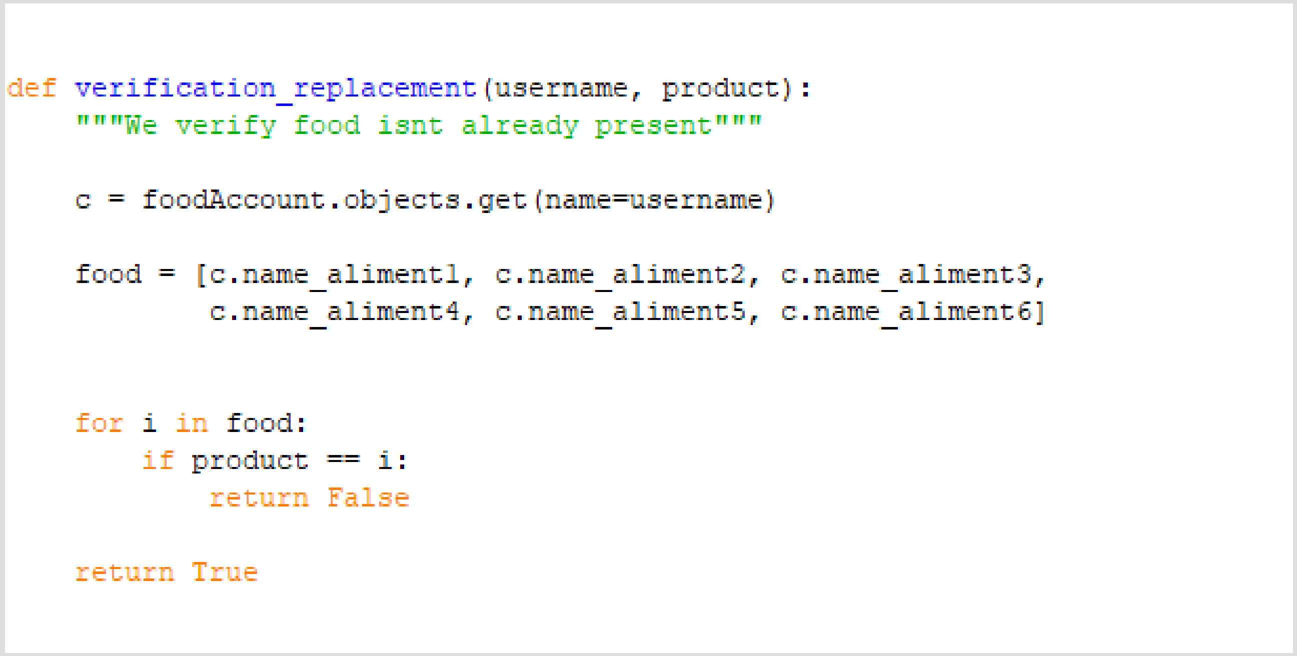
```
current_user = request.user
try:
    food = my_food_user(request.user.username)
    a = display_food(food)

    return render(request, 'mes_aliments.html', {"a":str(a[0][4]),
                                                  "b":str(a[1][4]),
                                                  "c":str(a[2][4]),
                                                  "d":str(a[3][4]),
                                                  "e":str(a[4][4]),
                                                  "f":str(a[5][4]),

                                                  "aa":str(a[0][0]),
                                                  "bb":str(a[1][0]),
                                                  "cc":str(a[2][0]),
                                                  "dd":str(a[3][0]),
                                                  "ee":str(a[4][0]),
                                                  "ff":str(a[5][0]),

                                                  "aaaa":"/static/img/portfolio/nutriscore/" + str(a[0][
                                                  "bbbb":"/static/img/portfolio/nutriscore/" + str(a[1][
                                                  "cccc":"/static/img/portfolio/nutriscore/" + str(a[2][
                                                  "dddd":"/static/img/portfolio/nutriscore/" + str(a[3][
                                                  "eeee":"/static/img/portfolio/nutriscore/" + str(a[4][
                                                  "ffff":"/static/img/portfolio/nutriscore/" + str(a[5][

                                                  "aaaaa":str(a[0][0]),
                                                  "bbbbb":str(a[1][0]),
                                                  "ccccc":str(a[2][0]),
                                                  "ddddd":str(a[3][0]),
                                                  "eeeee":str(a[4][0]),
                                                  "fffff":str(a[5][0]),
                                                  'message':message

                                                  })
except:
    return render(request, 'mes_aliments.html')
```

Vues - la fonction replacing



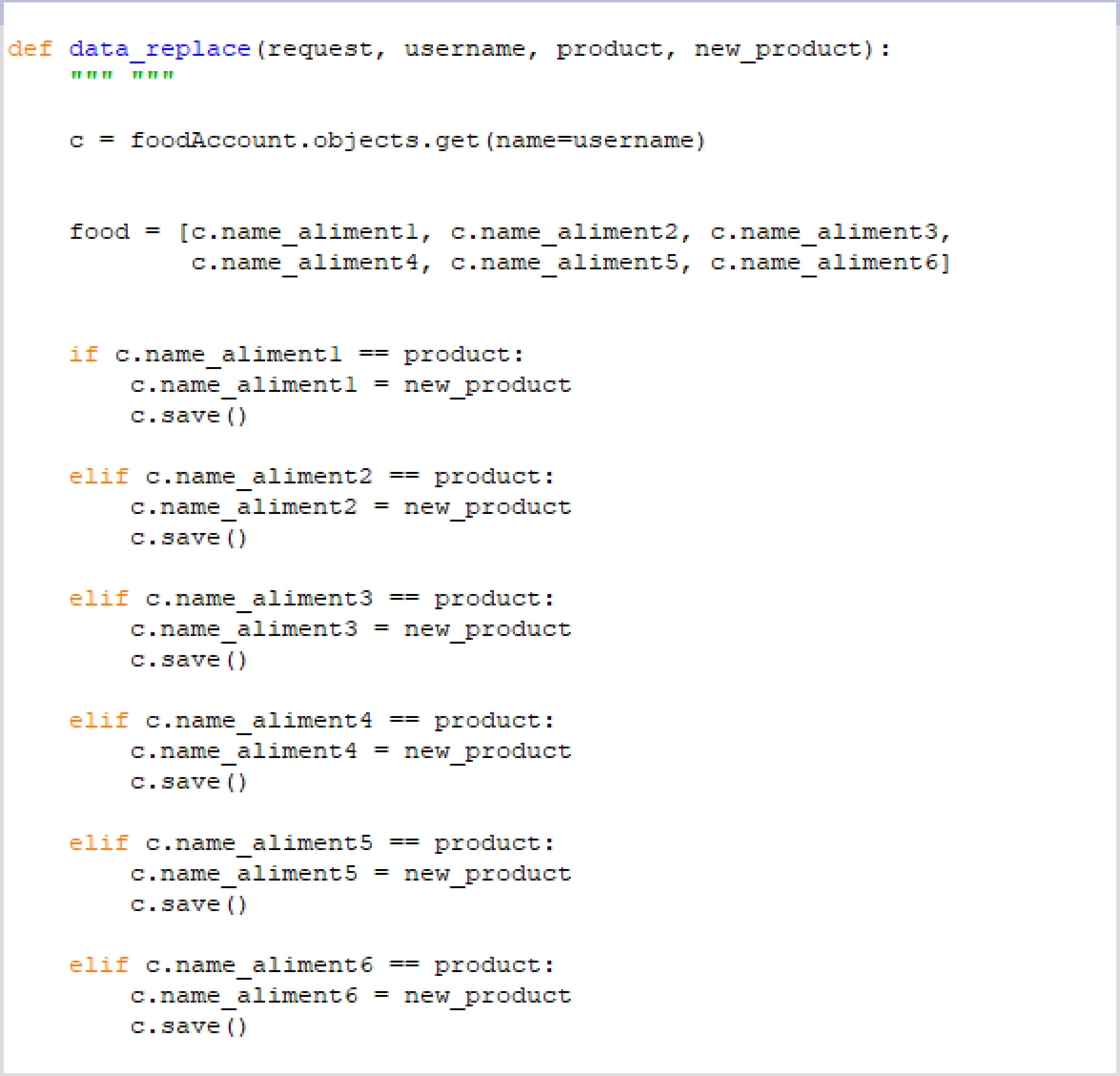
```python
def replacing(request):
    """This is functionality for replace food from my food"""


    message = ''

    if request.method == "POST":
        replace_it = request.POST.getlist('remplace_food')
        if replace_it:
            current_user = request.user

            liste = [[],[]]
            element = []
            c=0
            for i in replace_it:
                for j in i:
                    if j == ",":
                        c+=1
                    else:
                        liste[c].append(j)
                c+=1
            for i in liste:
                i = "".join(i)
                element.append(i)

            b = verification_replacement(current_user, "".join(liste[-1]))

            if b == True:
                data_replace(request, current_user,
                             element[0], element[1])
            elif b == False:
                message = 'vous avez deja cet aliment'
```

Vues - verification_replacement()



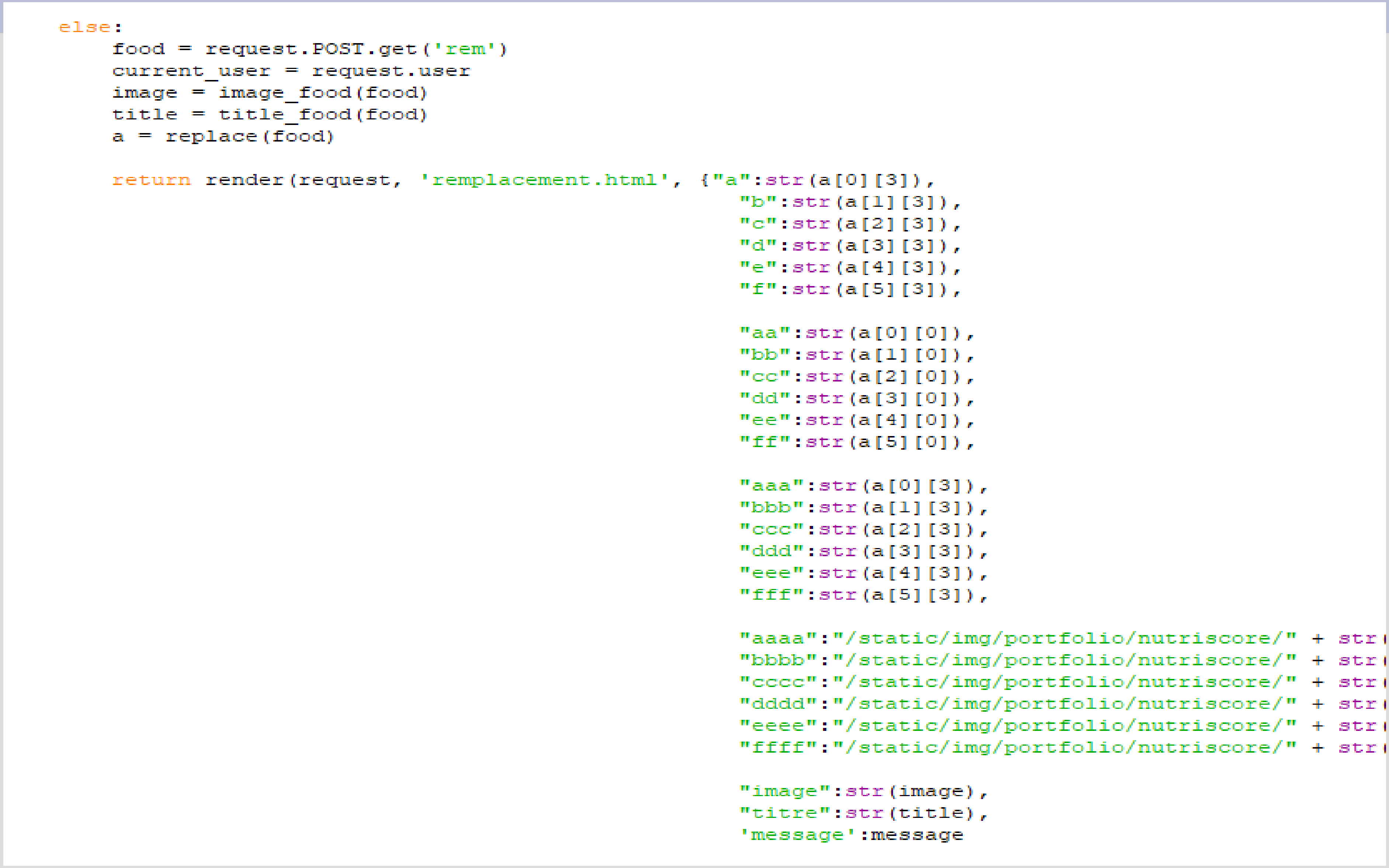
```python
def verification_replacement(username, product):
    """We verify food isnt already present"""

    c = foodAccount.objects.get(name=username)

    food = [c.name_aliment1, c.name_aliment2, c.name_aliment3,
            c.name_aliment4, c.name_aliment5, c.name_aliment6]


    for i in food:
        if product == i:
            return False

    return True
```

Vues - data_replace()



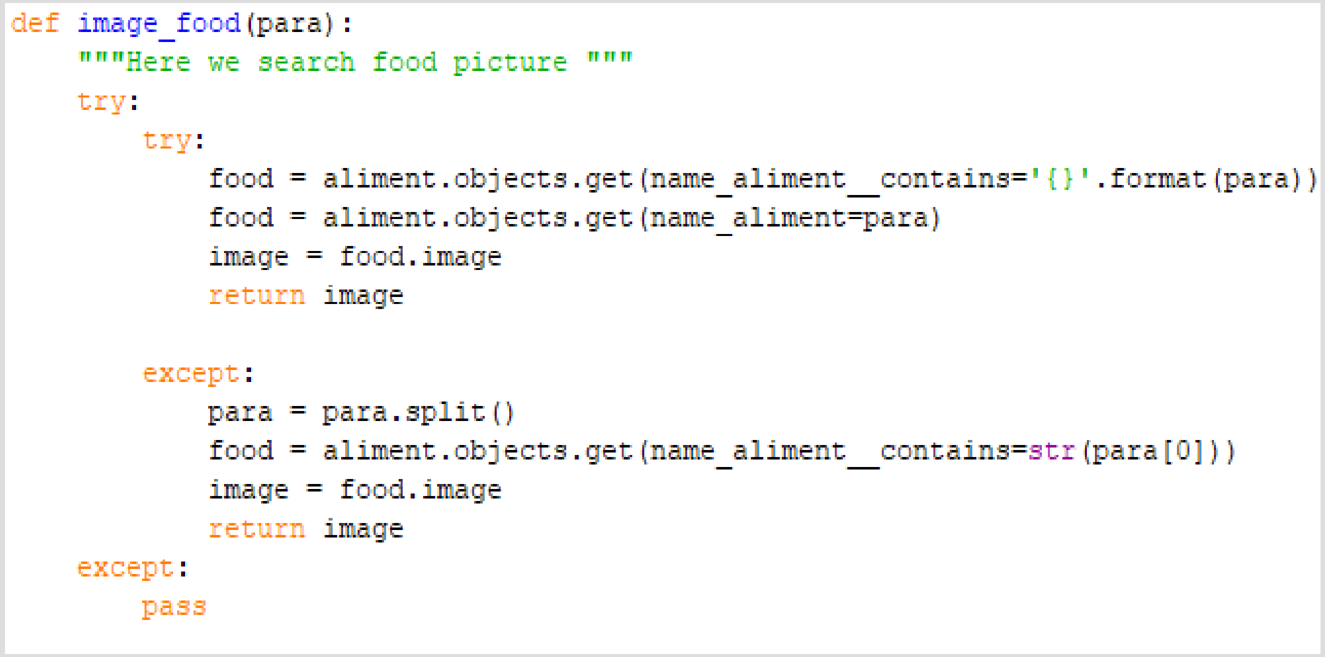
```
def data_replace(request, username, product, new_product):
    """ """

    c = foodAccount.objects.get(name=username)


    food = [c.name_aliment1, c.name_aliment2, c.name_aliment3,
            c.name_aliment4, c.name_aliment5, c.name_aliment6]


    if c.name_aliment1 == product:
        c.name_aliment1 = new_product
        c.save()

    elif c.name_aliment2 == product:
        c.name_aliment2 = new_product
        c.save()

    elif c.name_aliment3 == product:
        c.name_aliment3 = new_product
        c.save()

    elif c.name_aliment4 == product:
        c.name_aliment4 = new_product
        c.save()

    elif c.name_aliment5 == product:
        c.name_aliment5 = new_product
        c.save()

    elif c.name_aliment6 == product:
        c.name_aliment6 = new_product
        c.save()
```

Vues - la fonction replacing



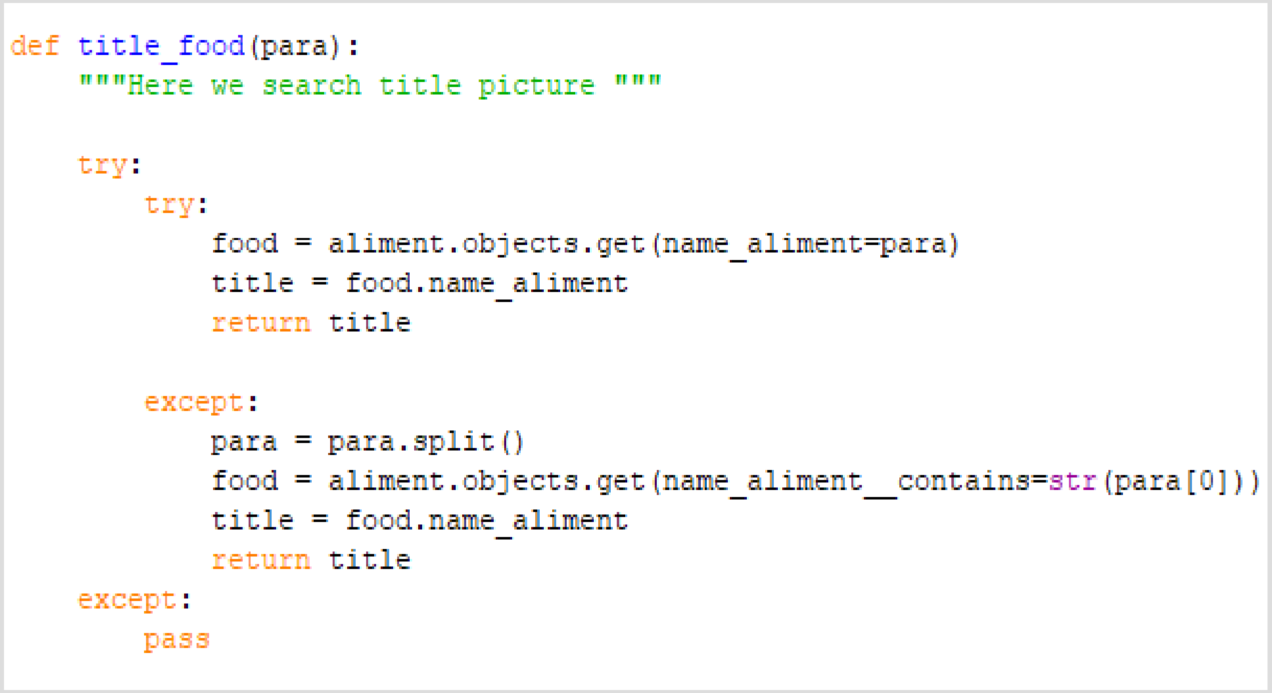
```
else:
    food = request.POST.get('rem')
    current_user = request.user
    image = image_food(food)
    title = title_food(food)
    a = replace(food)

    return render(request, 'remplacement.html', {"a":str(a[0][3]),
                                                   "b":str(a[1][3]),
                                                   "c":str(a[2][3]),
                                                   "d":str(a[3][3]),
                                                   "e":str(a[4][3]),
                                                   "f":str(a[5][3]),

                                                   "aa":str(a[0][0]),
                                                   "bb":str(a[1][0]),
                                                   "cc":str(a[2][0]),
                                                   "dd":str(a[3][0]),
                                                   "ee":str(a[4][0]),
                                                   "ff":str(a[5][0]),

                                                   "aaa":str(a[0][3]),
                                                   "bbb":str(a[1][3]),
                                                   "ccc":str(a[2][3]),
                                                   "ddd":str(a[3][3]),
                                                   "eee":str(a[4][3]),
                                                   "fff":str(a[5][3]),

                                                   "aaaa":"/static/img/portfolio/nutriscore/" + str
                                                   "bbbb":"/static/img/portfolio/nutriscore/" + str
                                                   "cccc":"/static/img/portfolio/nutriscore/" + str
                                                   "dddd":"/static/img/portfolio/nutriscore/" + str
                                                   "eeee":"/static/img/portfolio/nutriscore/" + str
                                                   "ffff":"/static/img/portfolio/nutriscore/" + str

                                                   "image":str(image),
                                                   "titre":str(title),
                                                   'message':message
```

Vues - image_food()



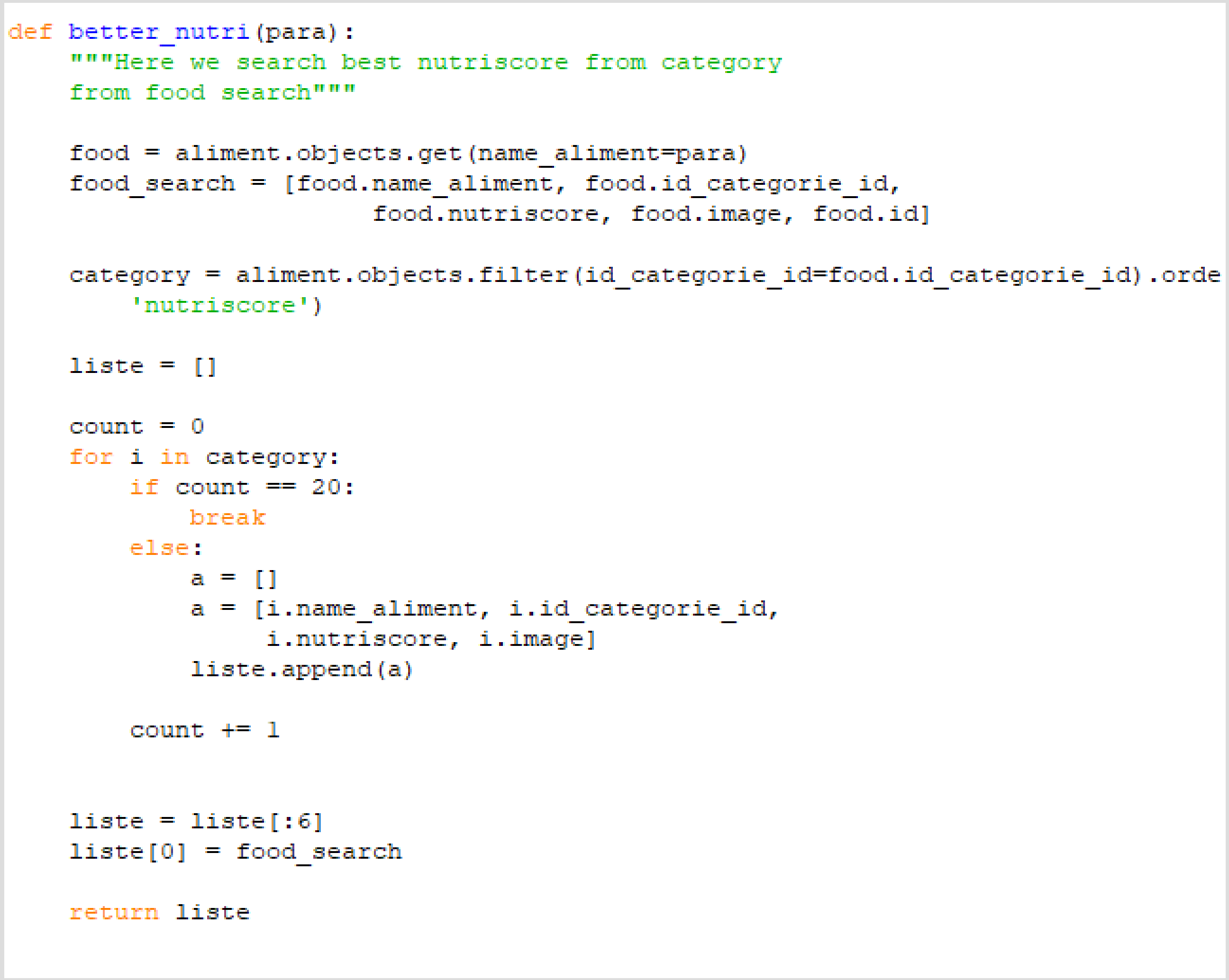
```python
def image_food(para):
    """Here we search food picture """
    try:
        try:
            food = aliment.objects.get(name_aliment__contains='{}'.format(para))
            food = aliment.objects.get(name_aliment=para)
            image = food.image
            return image

        except:
            para = para.split()
            food = aliment.objects.get(name_aliment__contains=str(para[0]))
            image = food.image
            return image
    except:
        pass
```

Vues - title_food()



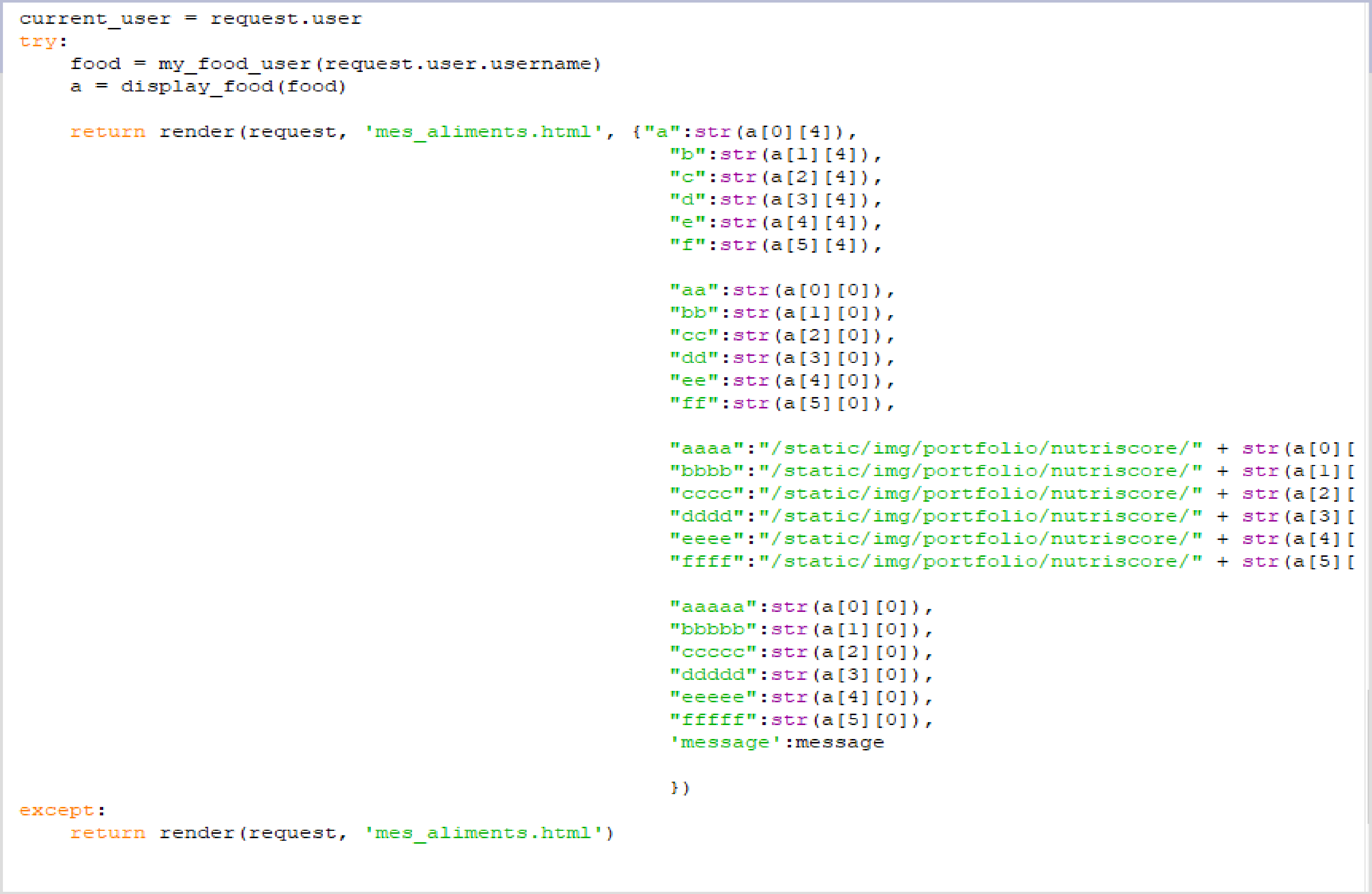
```python
def title_food(para):
    """Here we search title picture """

    try:
        try:
            food = aliment.objects.get(name_aliment=para)
            title = food.name_aliment
            return title

        except:
            para = para.split()
            food = aliment.objects.get(name_aliment__contains=str(para[0]))
            title = food.name_aliment
            return title
    except:
        pass
```

Vues - better_nutri()



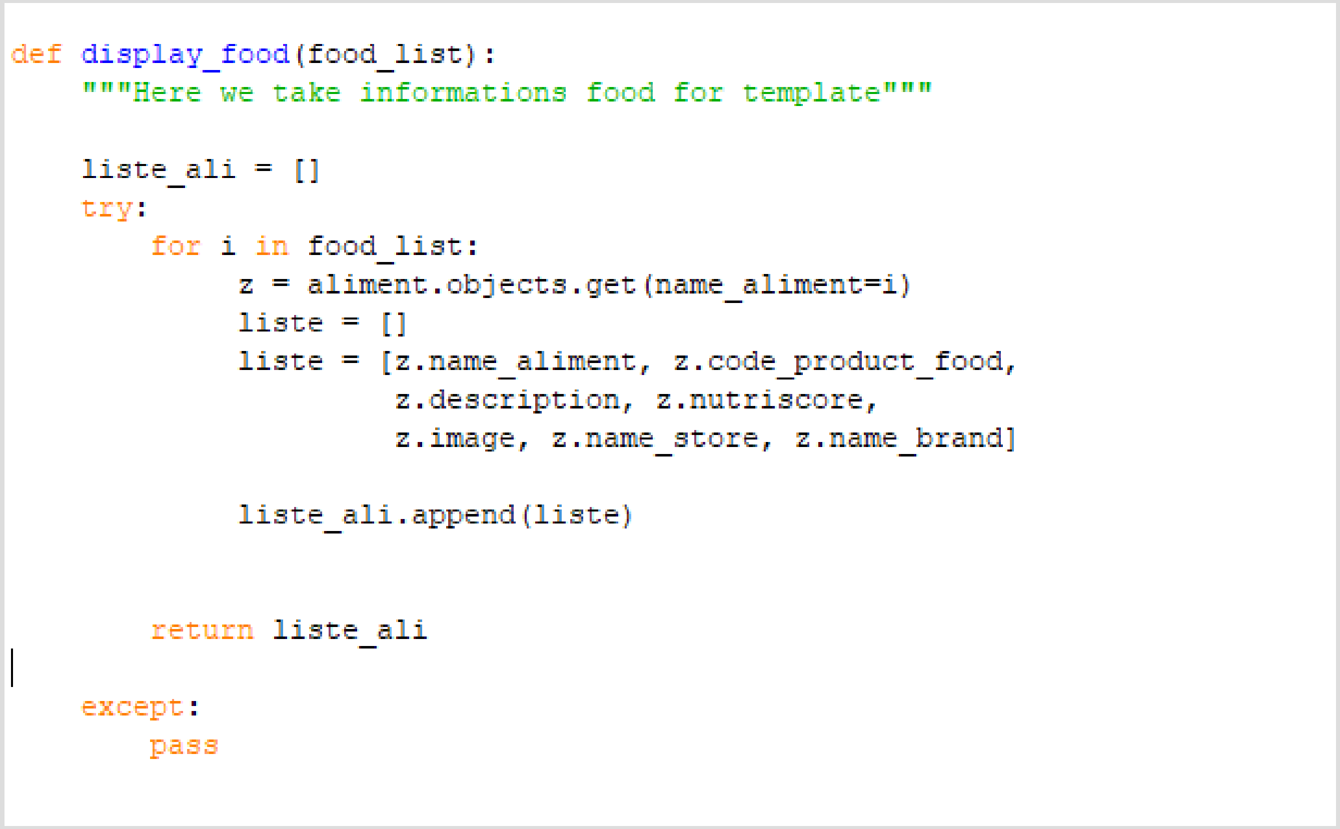
```python
def better_nutri(para):
    """Here we search best nutriscore from category
    from food search"""

    food = aliment.objects.get(name_aliment=para)
    food_search = [food.name_aliment, food.id_categorie_id,
                        food.nutriscore, food.image, food.id]

    category = aliment.objects.filter(id_categorie_id=food.id_categorie_id).orde
        'nutriscore')

    liste = []

    count = 0
    for i in category:
        if count == 20:
            break
        else:
            a = []
            a = [i.name_aliment, i.id_categorie_id,
                 i.nutriscore, i.image]
            liste.append(a)

        count += 1


    liste = liste[:6]
    liste[0] = food_search

    return liste
```

Vues - la fonction replacing



```
current_user = request.user
try:
    food = my_food_user(request.user.username)
    a = display_food(food)

    return render(request, 'mes_aliments.html', {"a":str(a[0][4]),
                                                   "b":str(a[1][4]),
                                                   "c":str(a[2][4]),
                                                   "d":str(a[3][4]),
                                                   "e":str(a[4][4]),
                                                   "f":str(a[5][4]),

                                                   "aa":str(a[0][0]),
                                                   "bb":str(a[1][0]),
                                                   "cc":str(a[2][0]),
                                                   "dd":str(a[3][0]),
                                                   "ee":str(a[4][0]),
                                                   "ff":str(a[5][0]),

                                                   "aaaa":"/static/img/portfolio/nutriscore/" + str(a[0][
                                                   "bbbb":"/static/img/portfolio/nutriscore/" + str(a[1][
                                                   "cccc":"/static/img/portfolio/nutriscore/" + str(a[2][
                                                   "dddd":"/static/img/portfolio/nutriscore/" + str(a[3][
                                                   "eeee":"/static/img/portfolio/nutriscore/" + str(a[4][
                                                   "ffff":"/static/img/portfolio/nutriscore/" + str(a[5][

                                                   "aaaaa":str(a[0][0]),
                                                   "bbbbb":str(a[1][0]),
                                                   "ccccc":str(a[2][0]),
                                                   "ddddd":str(a[3][0]),
                                                   "eeeee":str(a[4][0]),
                                                   "fffff":str(a[5][0]),
                                                   'message':message

                                                   })
except:
    return render(request, 'mes_aliments.html')
```

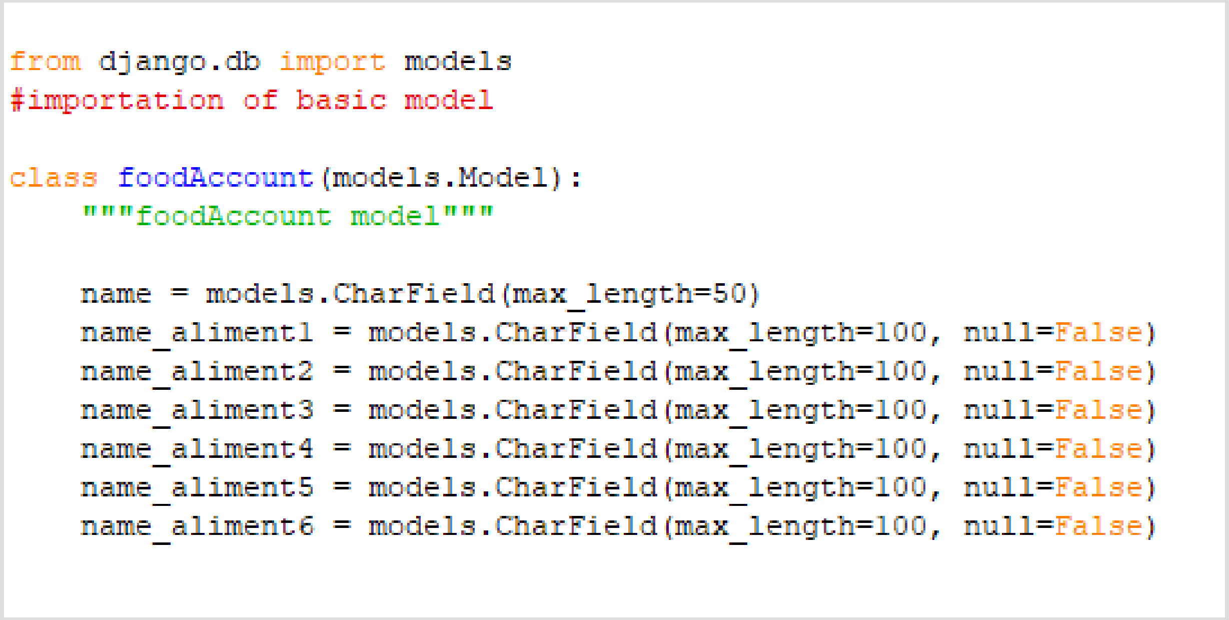
Vues - display_food()



```python
def display_food(food_list):
    """Here we take informations food for template"""

    liste_ali = []
    try:
        for i in food_list:
             z = aliment.objects.get(name_aliment=i)
             liste = []
             liste = [z.name_aliment, z.code_product_food,
                      z.description, z.nutriscore,
                      z.image, z.name_store, z.name_brand]

             liste_ali.append(liste)


        return liste_ali

    except:
        pass
```

Plan

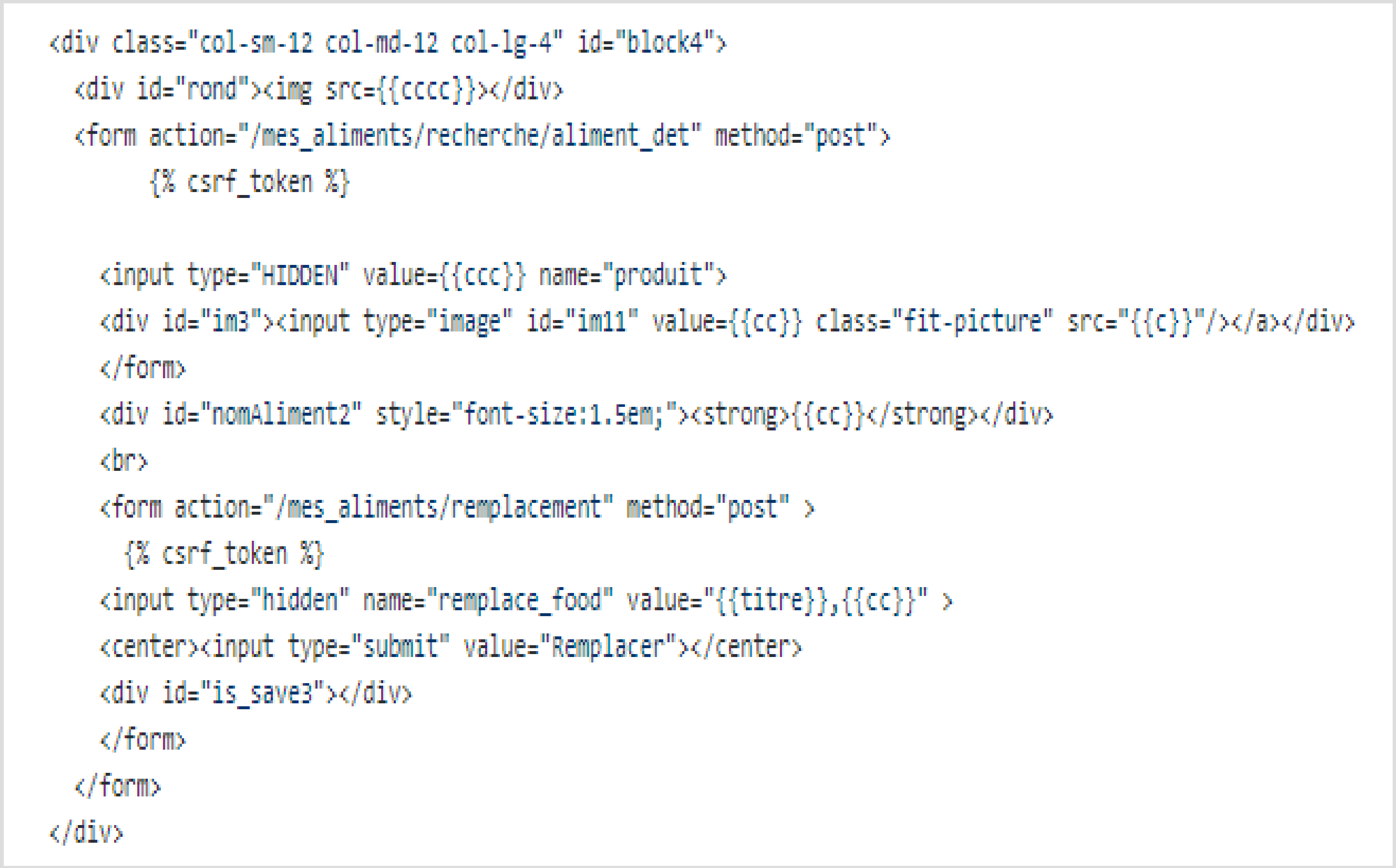
Model - foodAccount



```
from django.db import models
#importation of basic model

class foodAccount(models.Model):
    """foodAccount model"""

    name = models.CharField(max_length=50)
    name_aliment1 = models.CharField(max_length=100, null=False)
    name_aliment2 = models.CharField(max_length=100, null=False)
    name_aliment3 = models.CharField(max_length=100, null=False)
    name_aliment4 = models.CharField(max_length=100, null=False)
    name_aliment5 = models.CharField(max_length=100, null=False)
    name_aliment6 = models.CharField(max_length=100, null=False)
```

Plan

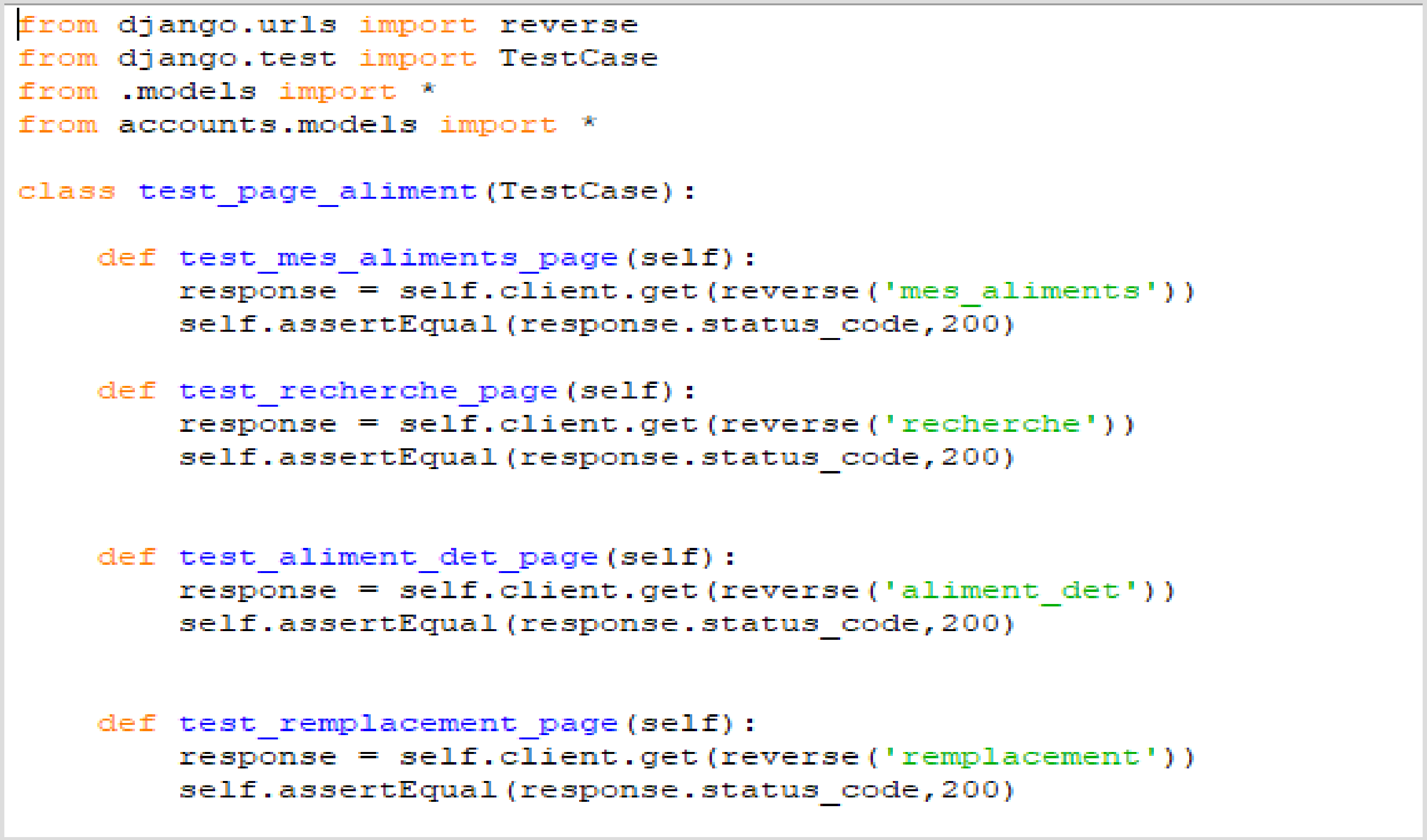
Template



```
<div class="col-sm-12 col-md-12 col-lg-4" id="block4">
  <div id="rond"><img src={{cccc}}></div>
  <form action="/mes_aliments/recherche/aliment_det" method="post">
        {% csrf_token %}

    <input type="HIDDEN" value={{ccc}} name="produit">
    <div id="im3"><input type="image" id="im11" value={{cc}} class="fit-picture" src="{{c}}"/></a></div>
    </form>
    <div id="nomAliment2" style="font-size:1.5em;"><strong>{{cc}}</strong></div>
    <br>
    <form action="/mes_aliments/remplacement" method="post" >
      {% csrf_token %}
    <input type="hidden" name="remplace_food" value="{{titre}},{{cc}}" >
    <center><input type="submit" value="Remplacer"></center>
    <div id="is_save3"></div>
    </form>
  </form>
</div>
```

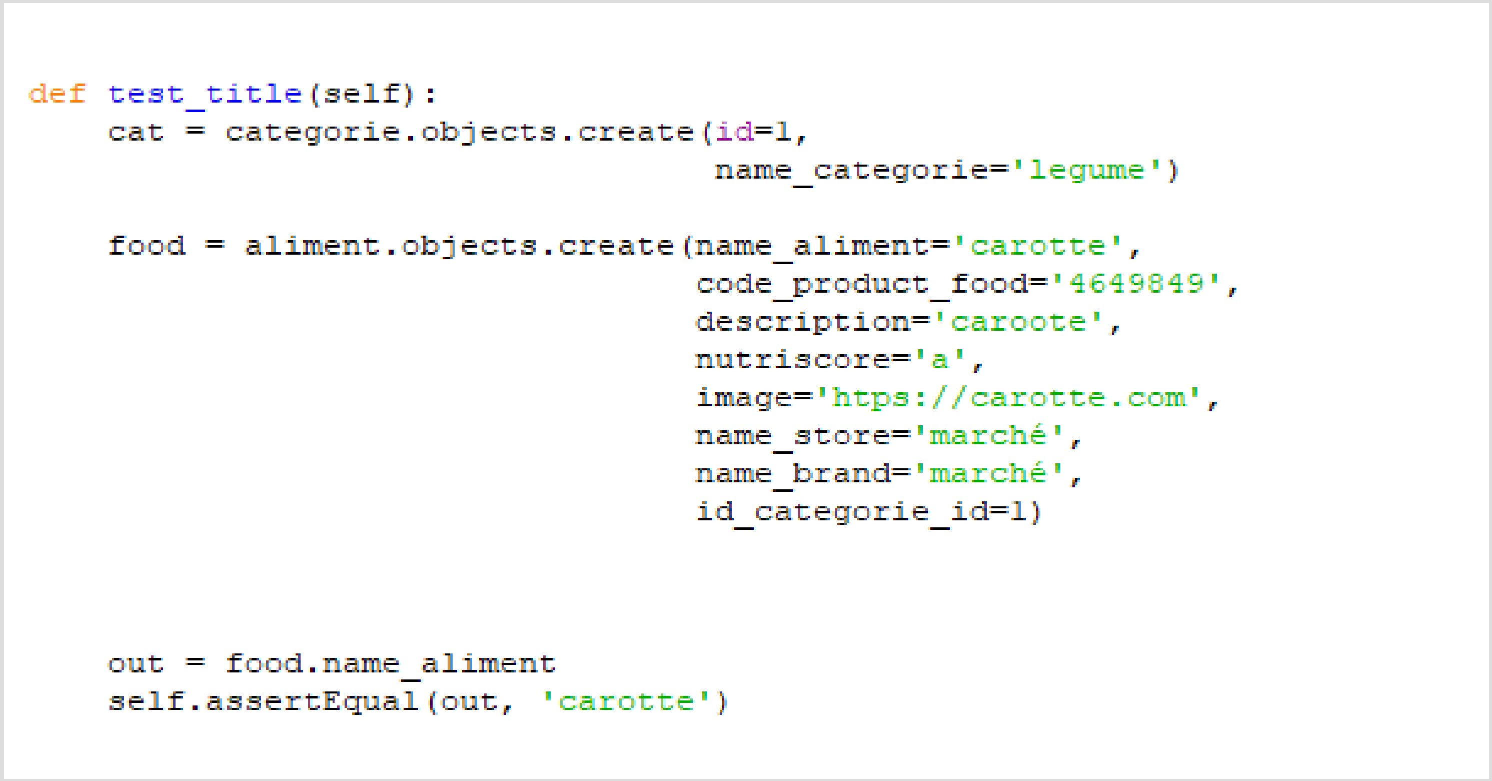
Plan

Test



```python
from django.urls import reverse
from django.test import TestCase
from .models import *
from accounts.models import *

class test_page_aliment(TestCase):

    def test_mes_aliments_page(self):
        response = self.client.get(reverse('mes_aliments'))
        self.assertEqual(response.status_code,200)

    def test_recherche_page(self):
        response = self.client.get(reverse('recherche'))
        self.assertEqual(response.status_code,200)


    def test_aliment_det_page(self):
        response = self.client.get(reverse('aliment_det'))
        self.assertEqual(response.status_code,200)


    def test_remplacement_page(self):
        response = self.client.get(reverse('remplacement'))
        self.assertEqual(response.status_code,200)
```

Test



```python
def test_title(self):
    cat = categorie.objects.create(id=1,
                                   name_categorie='legume')

    food = aliment.objects.create(name_aliment='carotte',
                                  code_product_food='4649849',
                                  description='caroote',
                                  nutriscore='a',
                                  image='htps://carotte.com',
                                  name_store='marché',
                                  name_brand='marché',
                                  id_categorie_id=1)



    out = food.name_aliment
    self.assertEqual(out, 'carotte')
```

Plan

V Intégration continue (HEROKU)

- Téléchargement Heroku Cli
- Création projet
- Création application
- Migrations
- Push

Plan

VI Gestion de projet

- Une méthode agile car :

  - définition des fonctions

  - début

  - estimation

  - fini le

  - commentaire

  - à travailler

- *Cf le trello*