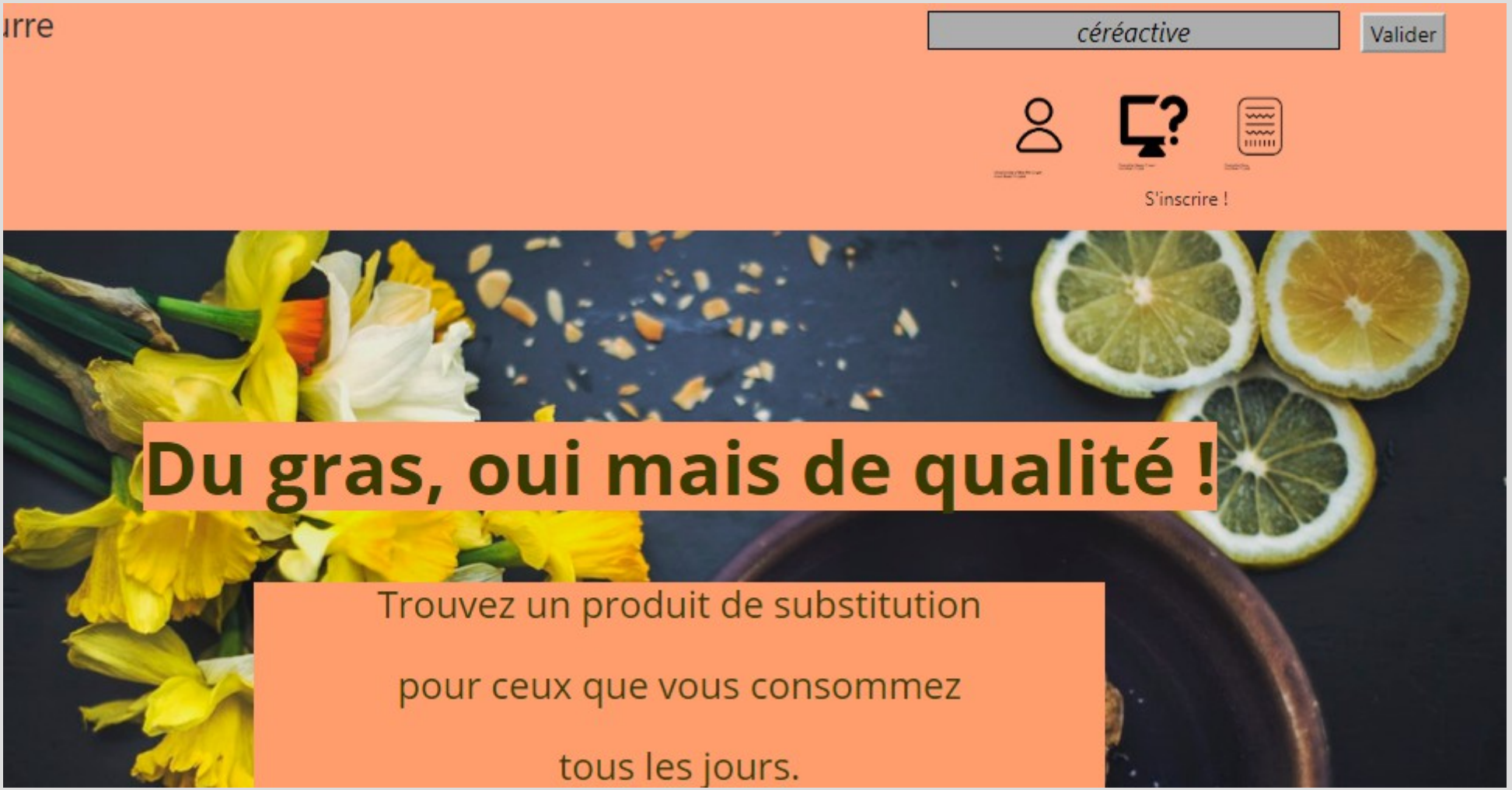
Projet 8

Plateforme nutella

Plateforme Nutella



Plan

I contexte

II Technologies utilisées

III Schéma de développement de Django

IV Cheminement complet d'une requête

V Intégration continue (HEROKU)

VI Gestion de projet

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

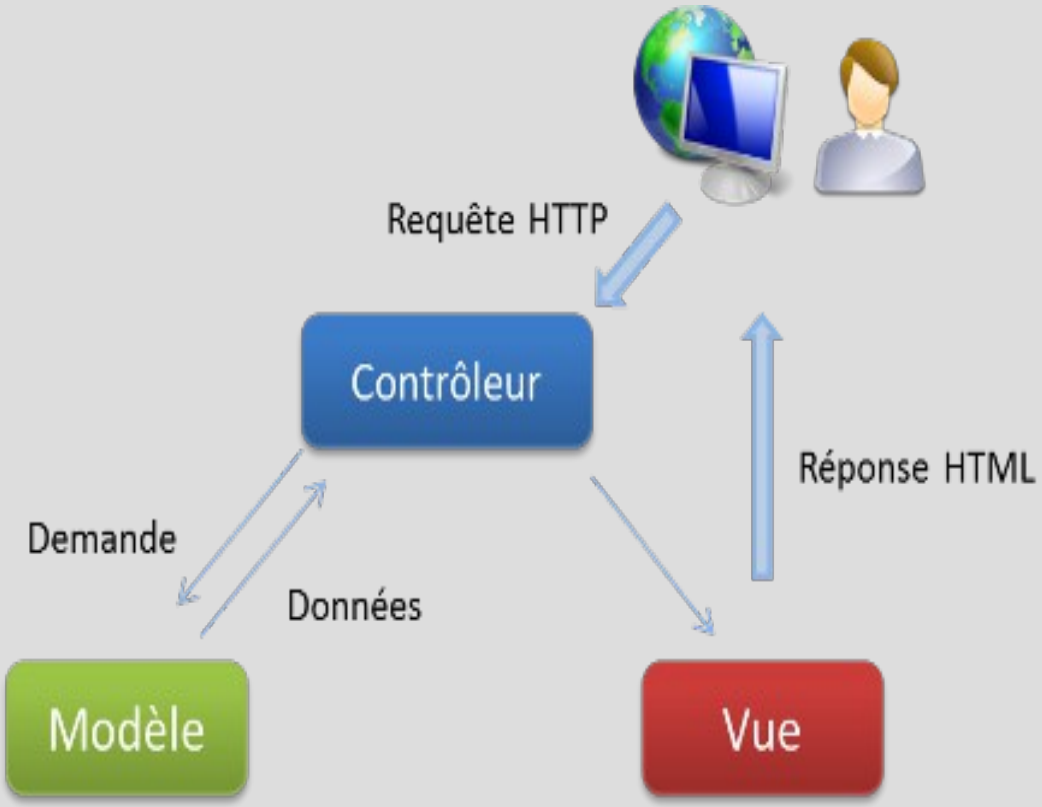
II Technologies utilisées

- différents langages de programmation

  - HTML/CSS *(affichage visuel)*
  - Javascript/Jquerry *(animation front et requête asyncrhone)*
  - Python *(Traitement de donnée)*
  - framework Django *(Traitement de donnée)*

III Schéma de développement de Django

Le modèle MVT



III Schéma de développement de Django

MODELE

- L'orm un outils pratique

  - gestion de la base de donnée

- *Donnée correspondant à une requête*
- *Réponse*

III Schéma de développement de Django

CONTROLEUR

- Les views

  - interaction ente les modèles et les templates

III Schéma de développement de Django

TEMPLATES

- Les templates
  - *restitution graphique générées par les vues*
- *Reçoit les requêtes http*
- *Renvoie une réponse*

IV Cheminement complet d'une requête

- Rooting

```
from django.contrib import admin
from django.urls import path
from .views import my_food
from .views import searching
from .views import food_det
from .views import replacing
from .views import error

urlpatterns = [
    path('/mes_aliments', my_food, name='mes_aliments'),
    path('/recherche', searching, name='recherche'),
    path('/aliment_det', food_det, name='aliment_det'),
    path('/remplacement', replacing, name='remplacement'),
    path('/error', error, name='error'),

    ]
```

Vues - la fonction replacing()

```
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()

```
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
```

Model - foodAccount

```python
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)
```

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>
```

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)
```

V Intégration continue (HEROKU)

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

VI Gestion de projet

- Une méthode agile car :

  - définition des fonctions

  - début

  - estimation

  - fini le

  - commentaire

  - à travailler

- *Cf le trello*