

Projet 8




Plateforme nutella

Contexte

irre

céréactive


Valider

S'inscrire !

Du gras, oui mais de qualité !

Trouvez un produit de substitution
pour ceux que vous consommez
tous les jours.



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

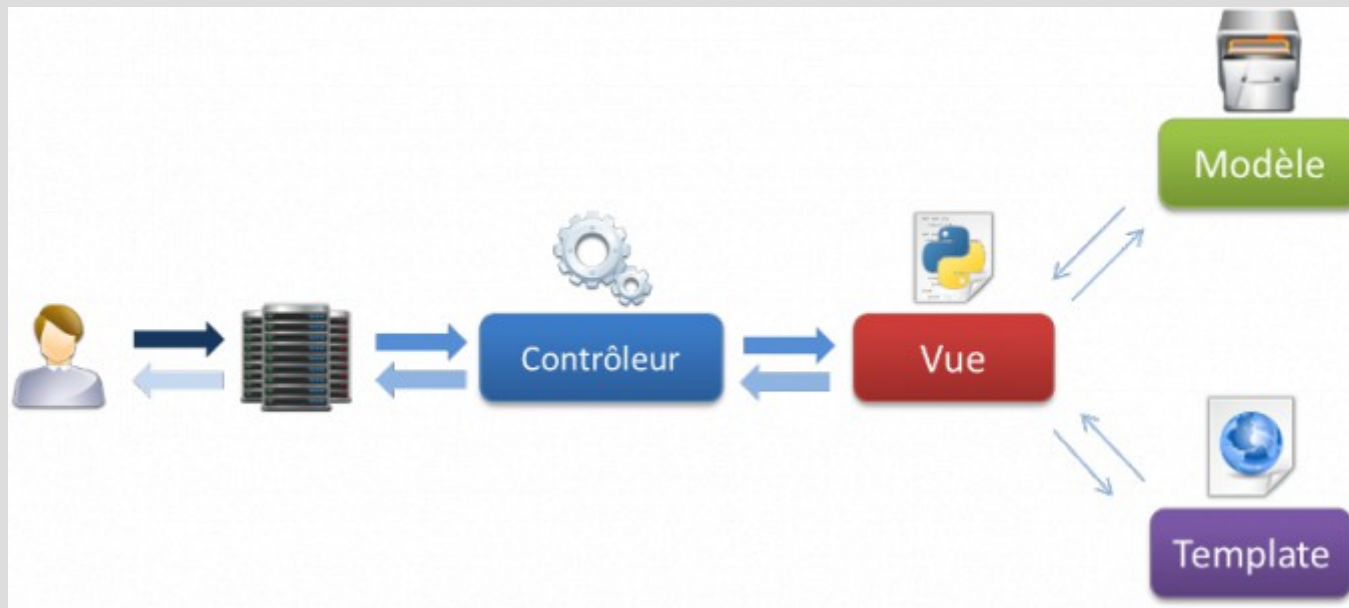
- Le site web
 - trouver des substituts plus sains à travers un site web.
- Plusieurs applications
 - connexion
 - recherche d'aliment
 - enregistrement
 - remplacement
 - visualisation

II Technologies utilisées

- différents langages de programmation
 - HTML/CSS
 - Javascript/Jquery
 - Python
 - framework Django

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

View

- Les views
 - interaction entre 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 food_det(request):  
    """This is function for food details"""  
  
    if request.method == "POST":  
  
        search = request.POST.get('produit')  
        details = food_details(search)  
        url_nutri = details.nutriscore  
        food = details.name_aliment  
        code = details.code_product_food  
        image = details.image  
        url_nutri = "/static/img/portfolio/nutriscore/" + str(url_nutri) + ".jpg >"  
  
        return render(request, 'aliment_det.html', {'detail': details,  
                                                    'url_nutri': url_nutri,  
                                                    'code': code,  
                                                    'image': image,  
                                                    'food': food  
                                                    })  
  
    return render(request, 'aliment_det.html')
```

Model - foodAccount

```
class aliment(models.Model):  
    """We create food table"""  
  
    name_aliment = models.CharField(max_length=100)  
    code_product_food = models.CharField(max_length=100)  
    description = models.TextField()  
    nutriscore = models.CharField(max_length=100)  
    image = models.ImageField()  
    name_store = models.CharField(max_length=100)  
    name_brand = models.CharField(max_length=100)  
    id_categorie = models.ForeignKey(categorie, on_delete=models.CASCADE)
```

Template

```
</div>
</header>
<section class="bg-primary" id="about">
  <div class="container">
    <div class="row">
      <div class="col-lg-8 mx-auto text-center">
        <div class="container-fluid">
          <div class="row" id="row3">
            <div class="col-md-6 col-xs-12" id="block10">
              <h1>Nutriscore<br> </h1>
            </div>
            <div class="col-md-6 col-xs-12" id="block12" >
              <h1>Repères nutritionnels<br>pour 100 g</h1>
            </div>
            <div class="col-md-12 col-xs-12" id="block13">
              <a href="https://fr.openfoodfacts.org/produit/{{code}}" target="_blank">
                <input type="submit" name="chrche1" value="Voir la fiche d'OpenFactFoods" id="button"/>
              </a>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
{% include "bottom_page.html" %}
```

Test

```
def test_image2():
    parametre = 'chocolat au lait'
    sortie = [('https://static.openfoodfacts.org/images/products/301/776/016.jpg', 'chocolat au lait')]
    assert script.image_aliment(parametre) == sortie

def test_titre_aliment():
    parametre = 'chocolat au lait'
    sortie = [('chocolat au lait',), ('chocolat au lait',), ('crêpes dentelles',)]
    assert script.titre_aliment(parametre) == sortie

def test_detail_aliment():
    parametre = 'https://static.openfoodfacts.org/images/products/327/016/016.jpg'
    sortie = [(22, '15 Mini Quiches', '3270160897087', 'Farine de blé_ (dur)',)]
    assert script.detail_aliment(parametre) == sortie

def test_verification_remplacement():
    sortie = False
    assert script.verification_remplacement('tete', 'chocolat au lait') == sortie
```

```
a\local\programs\python\python37-32\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\jeanbaptiste\plateforme_nutella\plateforme2\venv\plateforme, inifile:
collected 2 items

test_app_aliment_user.py::test_mes_aliment_user PASSED [ 50%]
test_app_aliment_user.py::test_display_food PASSED [100%]

===== 2 passed in 5.42 seconds =====

(venv) C:\Users\jeanbaptiste\plateforme_nutella\plateforme2\venv\plateforme>pytest test_app_recherche
.py -vv
===== test session starts =====
platform win32 -- Python 3.7.1, pytest-4.3.1, py-1.8.0, pluggy-0.9.0 -- c:\users\jeanbaptiste\appdata\local\programs\python\python37-32\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\jeanbaptiste\plateforme_nutella\plateforme2\venv\plateforme, inifile:
collected 5 items

test_app_recherche.py::test_image PASSED [ 20%]
test_app_recherche.py::test_image2 PASSED [ 40%]
test_app_recherche.py::test_titre_aliment PASSED [ 60%]
test_app_recherche.py::test_detail_aliment PASSED [ 80%]
test_app_recherche.py::test_verification_remplacement PASSED [100%]

===== 5 passed in 7.28 seconds =====

(venv) C:\Users\jeanbaptiste\plateforme_nutella\plateforme2\venv\plateforme>
```

1 Ko

V Intégration continue (HEROKU)

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

VI Gestion de projet

- Une méthode agile car :
 - explication méthode agile
- Cf le trello