



POLYTECH TOURS
64 avenue Jean Portalis
37200 TOURS, FRANCE
Tél +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Documents techniques

Site Web : location de produits électroniques

Tuteur académique :

Pascal Makris

Etudiants :

Rui Xu
Kuo Wang

1. Introduction générale

La tâche principale de ce projet est de développer un site Web pour la location de produits électroniques. Nous avons comparé quelques frameworks front-end et back-end existants, et avons finalement décidé d'utiliser Bootstrap pour le framework front-end et Django pour le framework back-end. Le langage de programmation correspondant à Django est Python.

Caractéristiques de Django :

- Fonctions complètes, livré avec un grand nombre d'outils et de cadres couramment utilisés (tels que la pagination, l'authentification, la gestion des autorités), adaptés au développement rapide de sites Web au niveau de l'entreprise.
- Documentation complète, après plus de dix ans de développement et d'amélioration, Django dispose d'un large éventail de cas pratiques et d'une documentation en ligne complète. Les développeurs peuvent rechercher des solutions dans des documents en ligne lorsqu'ils rencontrent des problèmes.
- Composants d'accès à la base de données, la couche Model de Django est livrée avec des composants ORM de base de données, permettant aux développeurs d'opérer sur la base de données sans apprendre SQL.
- Concept de conception de l'application de Django, l'application est enfichable, pas besoin, vous pouvez la supprimer directement, elle a peu d'effet sur le système global.
- Administrateur de système de gestion autonome, Une plate-forme complète de contrôle de gestion des données d'arrière-plan peut être réalisée avec seulement quelques lignes de configuration et de code.
- Les informations de débogage de Django sont détaillées, il est facile de savoir où le code est erroné.

Caractéristiques de Bootstrap :

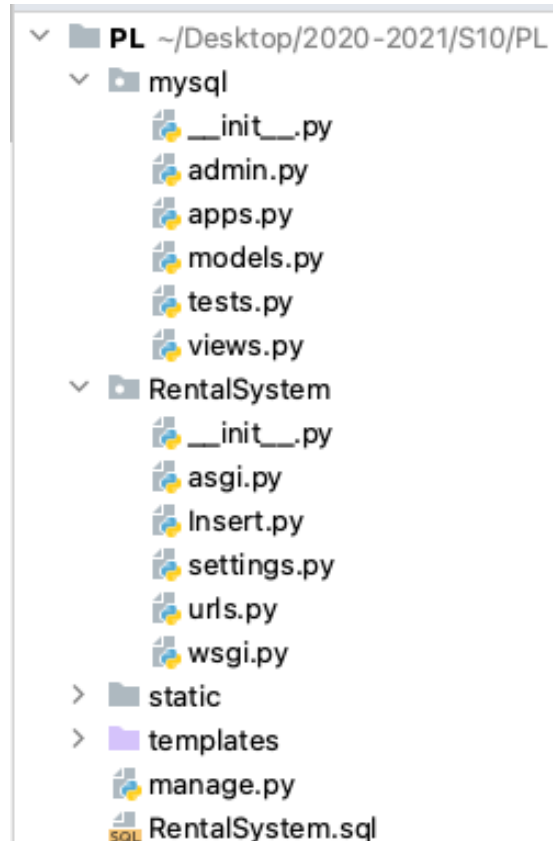
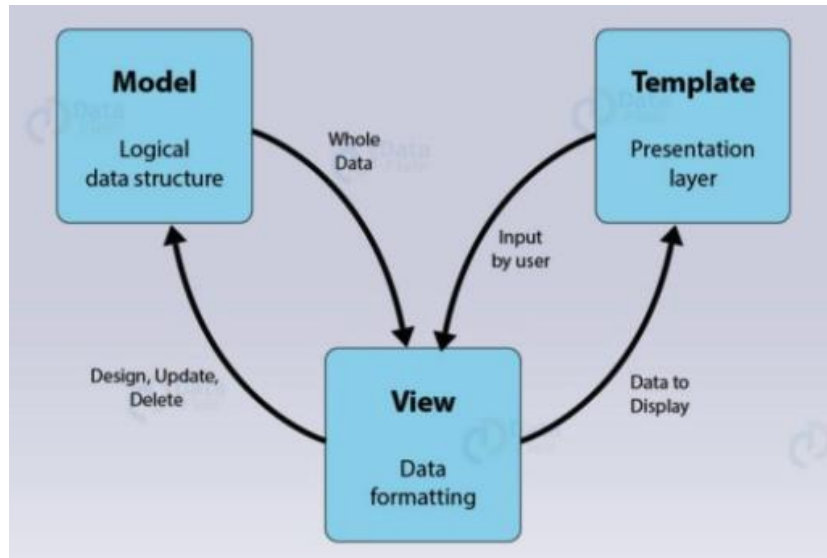
- Il a une forte évolutivité et peut mieux s'intégrer à de vrais projets de développement Web.
- Le cadre continue de s'adapter au développement de la technologie Web.
- Relativement mature, pleinement utilisé et testé dans un grand nombre de projets.
- Avec une documentation complète, il est plus pratique à utiliser.
- Bootstrap fournit un ensemble très riche de composants et de plug-ins, y compris de petites icônes, des groupes de boutons, une navigation dans les menus, des onglets, etc, et accepte la personnalisation.
- Extensibilité forte, compatible avec divers plug-ins de script.
- Avec des composants d'interface utilisateur prêts à l'emploi, vous pouvez créer rapidement des pages Web.
- Le framework Bootstrap fournit aux utilisateurs un système de grille de streaming réactif pour les appareils mobiles. Il a une structure de cadre complète et un effet global harmonieux. Il peut prendre en charge Google, Firefox, IE et d'autres navigateurs, et le développement du projet est pratique et rapide.

Dans ce document, la structure du projet et le manuel de développement sont principalement présentés.

2. Structure du projet

Django utilise le mode MTV :

- Modèle : responsable des objets métier et des objets de base de données (ORM).
- Template : responsable de la façon d'afficher la page à l'utilisateur.
- View : responsable de la logique métier, et appelle le modèle et Template le cas échéant.



manage.py : un outil en ligne de commande qui nous permet d'interagir avec les projets Django de différentes manières.

Dans RentalSystem :

- __init__.py: Un fichier vide, indiquant à python que ce répertoire est traité comme un package python (si vous avez ceci, vous pouvez l'appeler un package)
- settings.py: fichier de configuration du projet
- urls.py: déclaration d'URL du projet
- wsgi.py: Le projet est compatible avec l'entrée du serveur Web WSGI

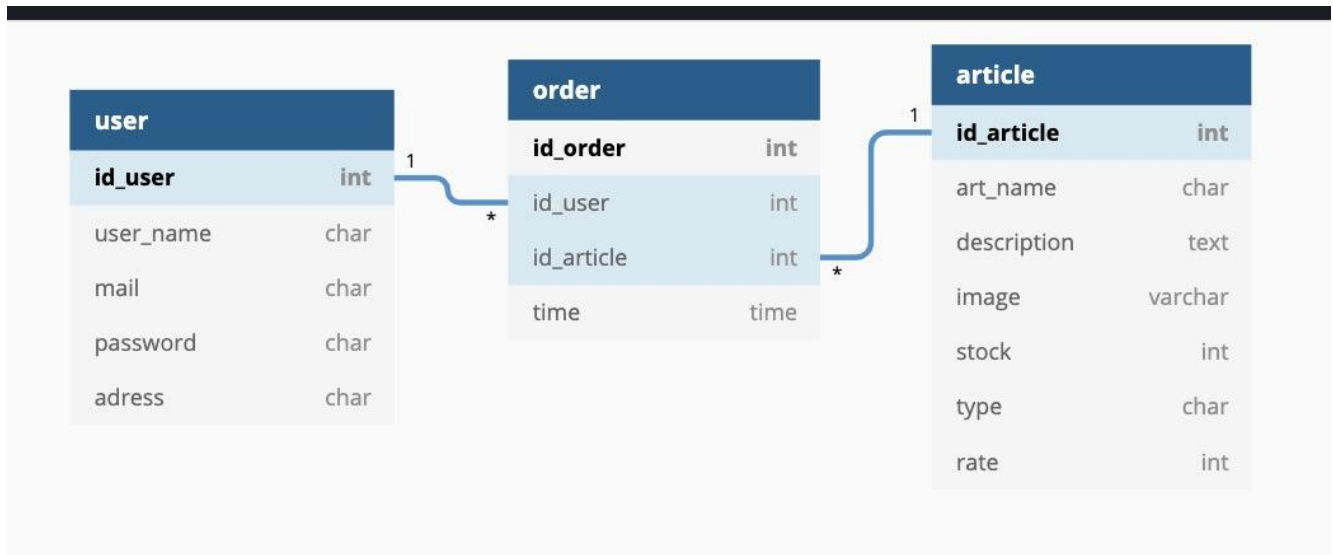
Dans mysql :

- admin.py: configuration du site
- app.py : configuration de l'application
- models.py: les modèles
- views.py: les vues
- tests.py : pour écrire des tests.

Dans le dossier templates, il y a des fichiers html de toutes les pages.

Dans le dossier static , nous pouvons stocker certains fichiers qui ne changeront pas, tels que les fichiers modèles css.

3. La bases de données



C'est la structure de la base de données.

Le site Web que nous avons implémenté n'est qu'un prototype et les fonctions implémentées sont relativement basiques, il n'y a donc pas trop de données et de tableaux conçus.

```

models.py x
1  from django.db import models
2
3  class User(models.Model):
4      user_name = models.CharField(max_length=32)
5      mail = models.EmailField()
6      password = models.CharField(max_length=128)
7      adress = models.TextField()
8      class Meta:
9          db_table = 'user'
10
11 class Article(models.Model):
12     art_name = models.CharField(max_length=32)
13     description = models.TextField()
14     image = models.CharField(max_length=256)
15     stock = models.IntegerField()
16     type = models.CharField(max_length=32)
17     rate = models.IntegerField()
18     class Meta:
19         db_table = 'article'
20
21 class Order(models.Model):
22     id_user = models.ForeignKey("User", on_delete=models.CASCADE)
23     id_article = models.ForeignKey("Article", on_delete=models.CASCADE)
24     time = models.DateTimeField(auto_now_add=True)
25     class Meta:
26         db_table = 'order'

```

Correspondant au code, dans le fichier models.py, nous devons créer les classes correspondantes, qui correspondent aux tables de la base de données une par une. Si la structure de la table ou les attributs de la base de données ont changé, vous devez le modifier ici.

Configuration de la base de données :

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'RentalSystem',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

```

Nous devons ajouter la configuration de la base de données dans le fichier setting.py.

Informations nécessaires : nom de la base de données, nom d'utilisateur, mot de passe, adresse d'accès, port.

4. Utilisation des vues

Dans Django, la vue répond à la requête WEB. L'essence de la vue est une fonction en python, qui est définie dans le fichier views.py.

La vue reçoit l'objet de requête comme premier paramètre, qui contient les informations demandées.

```
def login(request):
    if request.method == "POST":
        username = request.POST.get('username')
        password = hashlib.sha256(request.POST.get('pass').encode('utf-8')).hexdigest()
        res_user = User.objects.get(user_name=username)
        if password == res_user.password:
            request.session['is_login'] = 'True'
            request.session['username'] = username
            return redirect('/index/')
        else:
            return render(request, 'login.html')
    return render(request, 'login.html')
```

Ici, nous pouvons opérer sur les données frontales et les données de la base de données. Il est très pratique que vous n'ayez pas besoin d'utiliser SQL pour faire fonctionner la base de données.

5. Configuration du routage

```
import ...
urlpatterns = [
    path('admin/', admin.site.urls),
    path('insert/', Insert.insertUser),
    path('login/', views.login),
    path('index/', views.index),
    path('logout/', views.logout),
    path('detail/<Name>/', views.article, name='urlName'),
    path('rentals', views.rentals),
    path('register/', views.switchFunction),
    path('addUser/', views.addUser),
    path('changePass/', views.changePass),
    path('forgetPassword/', views.switchFunction),
    path('rent/<Name>/', views.rent, name='Name'),
    path('rent_art/<Name>/', views.rent_art, name='art'),
    path('profil/<user_name>', views.show_profil, name='user_name'),
    path('change/<user_name>', views.changePage, name='user_name'),
    path('', views.index)
```

Dans le fichier urls.py, ajoutez toutes les vues à cela et spécifiez quelle URL est donnée à quelle vue gérer.

6. Utilisation de templates

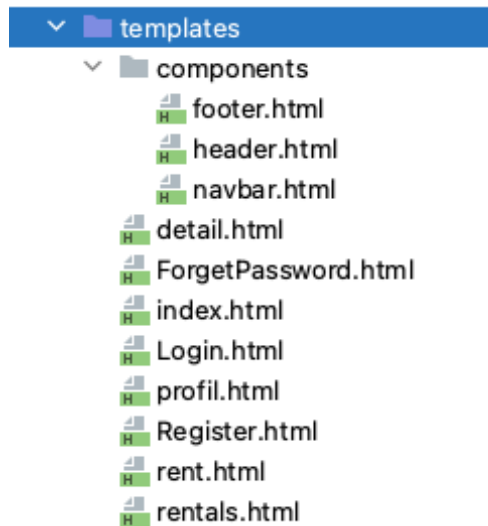
Dans le fichier settings.py, il y a la configuration de templates:

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

Ce sont toutes les pages que nous utilisons :



```

</ul>
{% if not request.session.is_login %}
<form class="d-flex" action="/login/" method="get">
<button class="btn btn-outline-success" type="submit">Login</button>
</form>
{% else %}
<form class="d-flex" action="/profil/{{request.session.username}}" method="get">
<button class="btn btn-outline-success" type="submit">{{ request.session.username }}</button>
</form>
<form class="d-flex" action="/logout/" method="get">
<button class="btn btn-outline-success" type="submit">Logout</button>
</form>
{% endif %}

```

Dans les templates, nous pouvons utiliser les données passées par les fonctions dans view.py.

7. Utilisation de Bootstrap

```

<html lang="zh-CN"><head>
  <title>EEE</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet">
  <link href="/static/styles/sticky-footer-navbar.css" rel="stylesheet">
  <script src="http://cdn.static.runoob.com/libs/jquery/2.1.1/jquery.min.js"></script>
  <script src="http://cdn.static.runoob.com/libs/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body class="d-flex flex-column h-100">

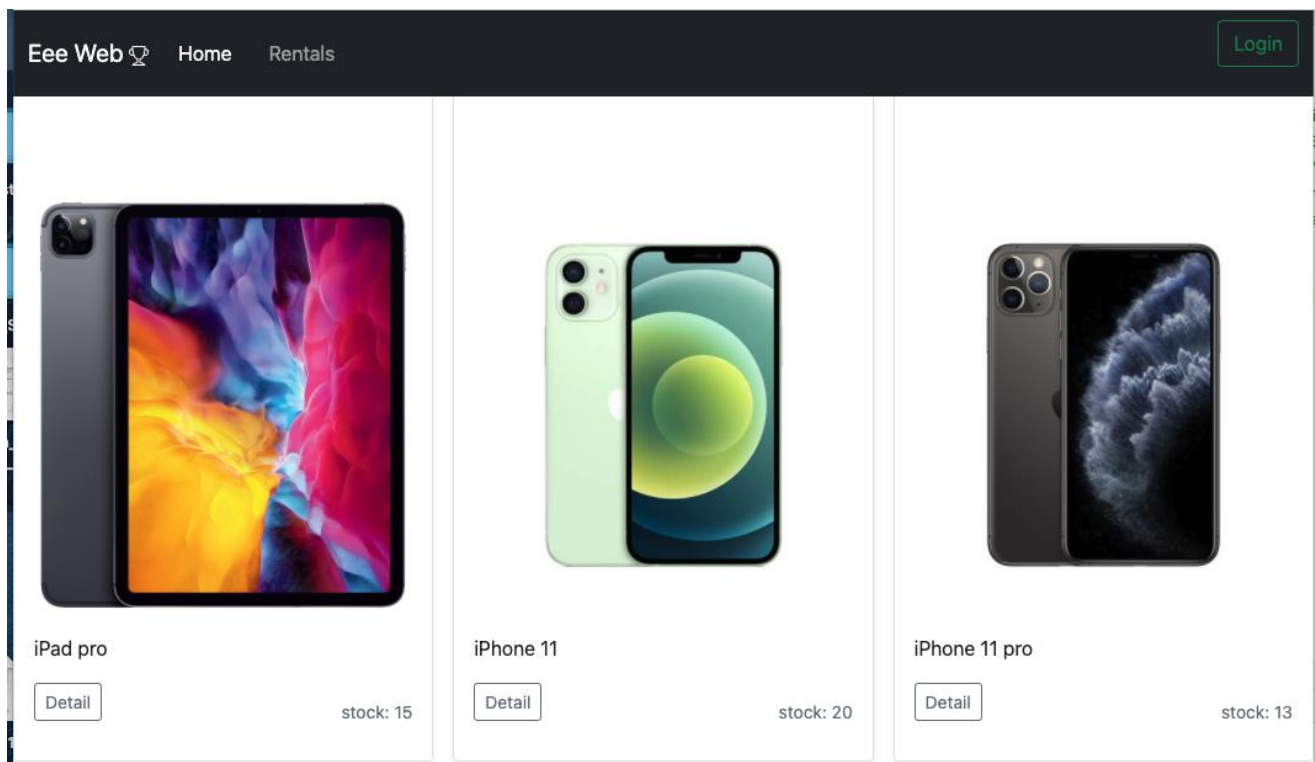
```

Listez les liens vers les fichiers de la bibliothèque Bootstrap que nous voulons utiliser dans l'en-tête de header.html.

La bibliothèque de Bootstrap est très riche, de nombreux composants peuvent être utilisés, ce qui suit est un exemple de l'interface que nous avons implémentée:

Page d'accueil :

Tous les produits électroniques seront affichés.



Page de location :

Lorsque l'utilisateur se connecte avec succès, il y a un bouton Louer pour chaque article, cliquez sur le bouton pour accéder à cette page, l'utilisateur peut choisir la durée de la location, puis passer une commande.

Si la commande est passée avec succès, il y aura une notification.

Eee Web Home Rentals smith Logout

Stock : 19
Per week : € 8
nb weeks: Value

Cancel Rent

Rent success! x

Page d'informations personnelles :

Une fois que l'utilisateur s'est connecté avec succès, cliquez sur le bouton correspondant au nom d'utilisateur dans le coin supérieur droit pour accéder à cette page, où vous pouvez voir les informations personnelles et l'historique des commandes.

Eee Web Home Rentals smith Logout

smith

Page d'accueil
Commande

Hello, smith

Nom d'utilisateur
smith

E-mail
smith@gmail.com

Adresse
grandmont batiment G, 37000 t

8. Commandes utilisées dans Django

Créer un nouveau projet :

```
django-admin.py startproject project_name
```

Créer une nouvelle application :

```
django-admin.py startapp app_name
```

Créer une table de base de données ou modifier une table ou un champ de base de données :

- Créez le fichier modifié : `python manage.py makemigrations`
- Appliquer le fichier py généré à la base de données : `python manage.py migrate`

Afin d'exécuter le serveur, nous pouvons passer la commande :

```
python manage.py runserver
```