

Compte Rendu :

# *Réalisation d'une application web avec DJANGO*



Enseignant : M. Stéphane TELETCHÉA

Réalisé par : Ismail ELFAQIR

Université Angers 2016- 2017  
Master 2 SILI

# PLAN

<b>Langages et logiciels utilisé</b>	<b>2</b>
<b>Analyse et conception</b>	<b>4</b>
1- Modèle conceptuel des données	4
2- Les modèles créé	5
<b>Configuration Django</b>	<b>7</b>
1- installations	7
2- pip list	7
3- configurations	8
4- Les fichiers .pyc	10
5- Tutoriels et étapes choisi	10
<b>Réalisation</b>	<b>11</b>
1- Page d'accueil	12
2- Shop page	13
3- Détail d'un produit	14
4- Gestion des produites	15
5- Panier	17
6- Contactez-nous	18
7- Administration	18

# Langages et logiciels utilisé

Dans cette partie je vais détailler les différents outils (langages et logiciels) utilisés pour la réalisation de ce projet :

**HTML/CSS** : (L'Hypertext Markup Language/Cascading Style Sheets), généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte. Le CSS (feuilles de style en cascade) est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, comprennent du code qui permet de gérer le design d'une page en HTML.

**Bootstrap** : est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble d'outils contenant des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

**Javascript** : est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web.

**AJAX** : L'architecture informatique Ajax (acronyme d'Asynchronous Javascript and Xml) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.

**JQuery** : des bibliothèques JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.

**Django** : un framework open-source de développement web en Python. Il a pour but de rendre le développement web 2.0 simple et rapide. Pour cette raison, le projet a

pour slogan « Le framework web pour les perfectionnistes sous pression ». Développé en 2003 pour le journal local de Lawrence (Kansas), Django a été publié sous licence BSD à partir de juillet 2005.

**MySQL** : est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde<sup>2</sup>, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

**Sublime Text** : un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.

Depuis la version 2.0, sortie le 26 juin 2012<sup>2</sup>, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langues plus rares.



# Analyse et conception

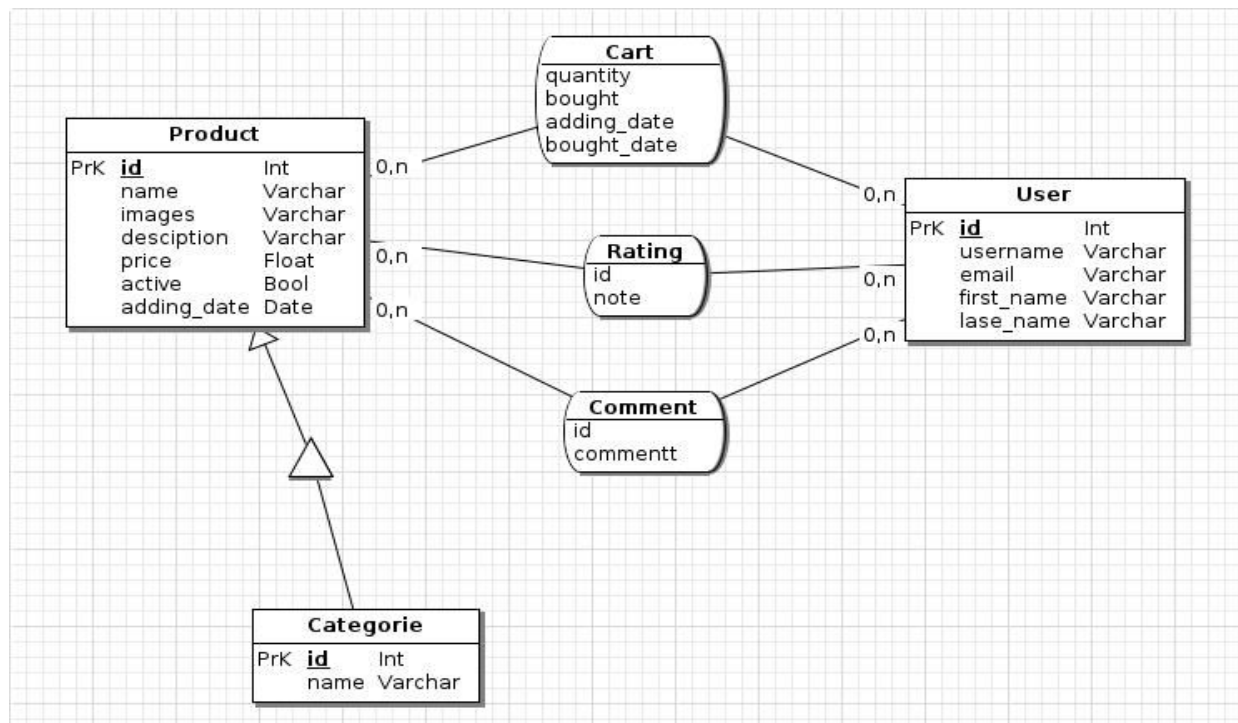
Cette partie est consacrée à la conception et la modélisation qui sont deux étapes fondamentales pour le développement d'une application web.

## 1- Modèle conceptuel des données

Le modèle conceptuel des données a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible, permettant de décrire le système d'informations à l'aide d'entités.

Dans mon cas, la description du MCD:

- Un utilisateur peut ajouter un ou plusieurs produits
- Un produit peut être ajouter par un seul utilisateur.
- L'utilisateur peut ajouter un ou plusieurs commentaire sur un produit
- L'utilisateur peut ajouter une note sur un produit
- Un produit peut être caractérisé par une seule catégorie
- Une catégorie peut contenir un ou plusieurs produits



## 2- Les modèles créé

- la Catégorie :

```
class Catégorie(models.Model):
    name = models.CharField(max_length=30)
    def __str__(self):
        return self.name
```

- Le Produit :
  - le 'owner' propriétaire du produit de type user
  - 'categorie' la catégorie du produit
  - 'active' pour vérifier si le produit est supprimé par son propriétaire
  - 'images' un produit peut avoir plusieurs images

```
class Product(models.Model):
    owner = models.ForeignKey(User)
    name = models.CharField(max_length=255)
    images = models.TextField(null=True)
    description = models.TextField(null=True)
    price = models.DecimalField(max_digits=12, decimal_places=2)
    adding_date = models.DateTimeField(auto_now_add=True)
    active = models.BooleanField(default=True)
    categorie = models.ForeignKey('Catégorie')
    def __str__(self):
        return "Product {0}".format(self.name)
```

- La Panier :
  - user : est une clé étrangère vers la table User (l'utilisateur qui a ajouté le produit au panier)
  - product : le produit ajouté au panier
  - quantity: la quantité
  - bought : attribut booléen pour vérifier si le produit est acheté

```
class Cart(models.Model):
    user = models.ForeignKey(User)
    product = models.ForeignKey(Product)
    quantity= models.IntegerField(default=1)
    bought = models.BooleanField(default=False)
    adding_date=models.DateTimeField(auto_now_add=True)
    bought_date=models.DateTimeField(auto_now=True)
    def __str__(self):
        return self.user.username
```

- Noter un produit :
  - user: l'utilisateur qui a noté le produit
  - product: le produit noté
  - like: la note /10

```
class RatingProduct(models.Model):  
    product = models.ForeignKey(Product)  
    user = models.ForeignKey(User)  
    like = models.IntegerField(default=0)  
    def __str__(self):  
        return self.product.name
```

- commenter un produit :
  - user: l'utilisateur qui a commenté le produit
  - product : le produit commenté
  - comment: le commentaire

```
class CommentProduct(models.Model):  
    product = models.ForeignKey(Product)  
    user = models.ForeignKey(User)  
    comment = models.TextField()  
    adding_date=models.DateTimeField(auto_now_add=True)  
    def __str__(self):  
        return self.product.name
```

# Configuration Django

## 1- installations

- ❖ Pour commencer j'ai installé MySQL en exécutant les commandes suivantes :

```
sudo apt-get install mysql-client  
sudo apt-get install mysql-server
```

- ❖ Des libs nécessaire pour django

```
sudo apt-get install python-virtualenv python-pip  
libmysqlclient-dev python-dev
```

- ❖ Après j'ai créé mon environnement virtuel en exécutant la commande suivante:

```
virtualenv ismail_elfaqir  
et pour le lancer  
source bin/activate
```

- ❖ Et puis en utilisant **python-pip** j'ai installer les suivants :

```
pip install django==1.8.17  
pip install MySQL-python  
pip install django-mathfilters  
pip install Pillow
```

## 2- pip list

En exécutant la commande **pip list** sur mon environnement virtuel il me donne la liste des pip installées :

**appdirs** : Un petit module Python pour déterminer des dirs appropriés spécifiques à une plate-forme, par ex. Un "répertoire de données utilisateur".

**Django**: framework Djnago version 1.8.17

**django-mathfilters** : des filters pour faire des calcules simple sur la template(vue)  
par exemple `{{ 10|dev:2 }}` = 5

**MySQL-python** : Pour connecter à mysql

**olefile** : pour lire et écrire les fichiers (Structured Storage)

**packaging** : Utilitaires de base pour les packages Python



**Pillow** : Django se sert en effet de cette dernière pour faire ses traitements sur les images

**pip** : (installing Python packages)


**pkg-resources**: Détection des paquets et accès aux ressources

**pyparsing** : Un module d'analyse générale pour Python

**setuptools**: est une bibliothèque complète, activement maintenue et stable conçue pour faciliter l'intégration des projets Python

**six**: fournit des utilitaires simples pour envelopper les différences entre Python 2 et Python 3

**wheel**: sont le nouveau standard de distribution des pythons et sont destinées à remplacer les eggs



```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
(ismail_elfaqir) $[etudiant@GBZ]22:54:15:~/Bureau/ismail/ismail_elfaqir/ecommerce } pip list
DEPRECATION: The default format will switch to columns in the future. You can use --format=(legacy|columns) (or define a format=(legacy|columns) in your pip.conf under the [list] section) to disable this warning.
appdirs (1.4.0)
Django (1.8.17)
django-mathfilters (0.4.0)
MySQL-python (1.2.5)
olefile (0.44)
packaging (16.8)
Pillow (4.0.0)
pip (9.0.1)
pkg-resources (0.0.0)
pyparsing (2.1.10)
setuptools (34.2.0)
six (1.10.0)
wheel (0.30.0a0)
(ismail_elfaqir) $[etudiant@GBZ]22:54:19:~/Bureau/ismail/ismail_elfaqir/ecommerce }
```

### 3- configurations

Le fichier de configuration (settings.py) est générée automatiquement après la création d'un projet Django par la commande suivante:

**django-admin.py startproject ecommerce**

- En premier temps j'ai modifier la configuration de la base de données en modifiant les lignes suivantes sur (settings.py)

```
DATABASES = {
'default': {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'ecommerce',
    'USER': 'root',
    'PASSWORD': 'ce1mdpp',
    'HOST': '',
    'PORT': '',
}
...

```

- J'ai ajouté une app par la commande suivante :  
**python manage.py startapp client**  
et je l'ai ajouté sur le fichier (settings.py) avec '**mathfilters**' que j'ai installer avec pip

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    ''mathfilters'',
    ''client'',
)

```

- J'ai ajouté un dossier pour les templates générale de mon application comme le HTML de base, la page login ...

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            os.path.join(BASE_DIR, "templates"),
        ], ...
    }
]

```

- pour les fichier static j'ai créé un répertoire (static) dans la racine de mon projet qui contient tous les fichier CSS et Js et d'autre fichier nécessaire pour l'application et pour l'inclure j'ai ajoutée les lignes suivantes

```
STATICFILES_DIRS = (  
    os.path.join(BASE_DIR, "static"),  
)
```

- Pour les images des produits télécharger j'ai créé un répertoire files sur le dossier static et pour l'inclure :

```
MEDIA_URL = '/media/'  
MEDIA_ROOT = os.path.join(BASE_DIR, 'static/files')
```

- pour envoyer des email à l'administrateur en utilisant un compte gmail

```
EMAIL_USE_TLS = True  
EMAIL_HOST = 'smtp.gmail.com'  
EMAIL_HOST_USER = 'parsimony.score@gmail.com'  
EMAIL_HOST_PASSWORD = 'parsimony.123456'  
EMAIL_PORT = 587
```

## 4- Les fichiers .pyc

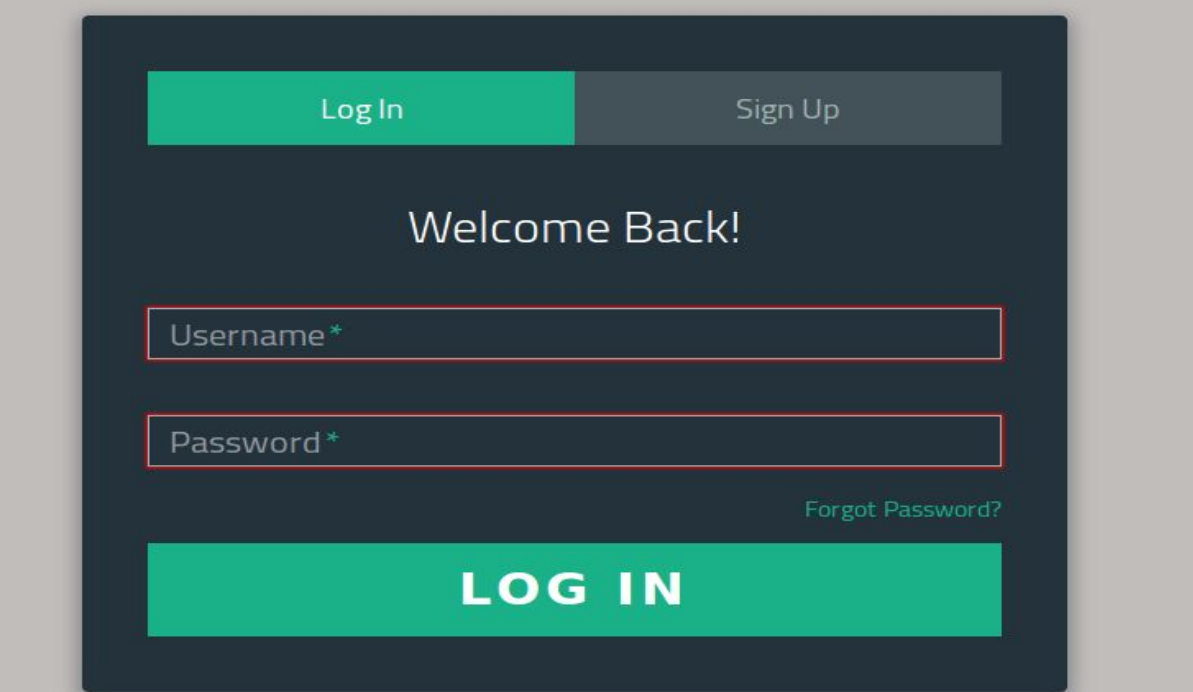
Les fichiers .pyc présents dans le projet correspondent à des fichiers .py compilé. Ce qui explique pourquoi nous n'avons pas de fichiers .pyc quand nous créons un projet ou un module django pour la première fois. Python va regarder en premier les fichiers .pyc et détecte les changements ce qui lui permet de ne pas re-compiler systématiquement les fichiers .py et donc gagner du temps et être plus rapide.

## 5- Tutoriels et étapes choisi

- <https://openclassrooms.com/courses/developpez-votre-site-web-avec-le-framework-django/creez-vos-applications-web-avec-django>
- <https://www.djangoproject.com/>
- Le diapo du TP

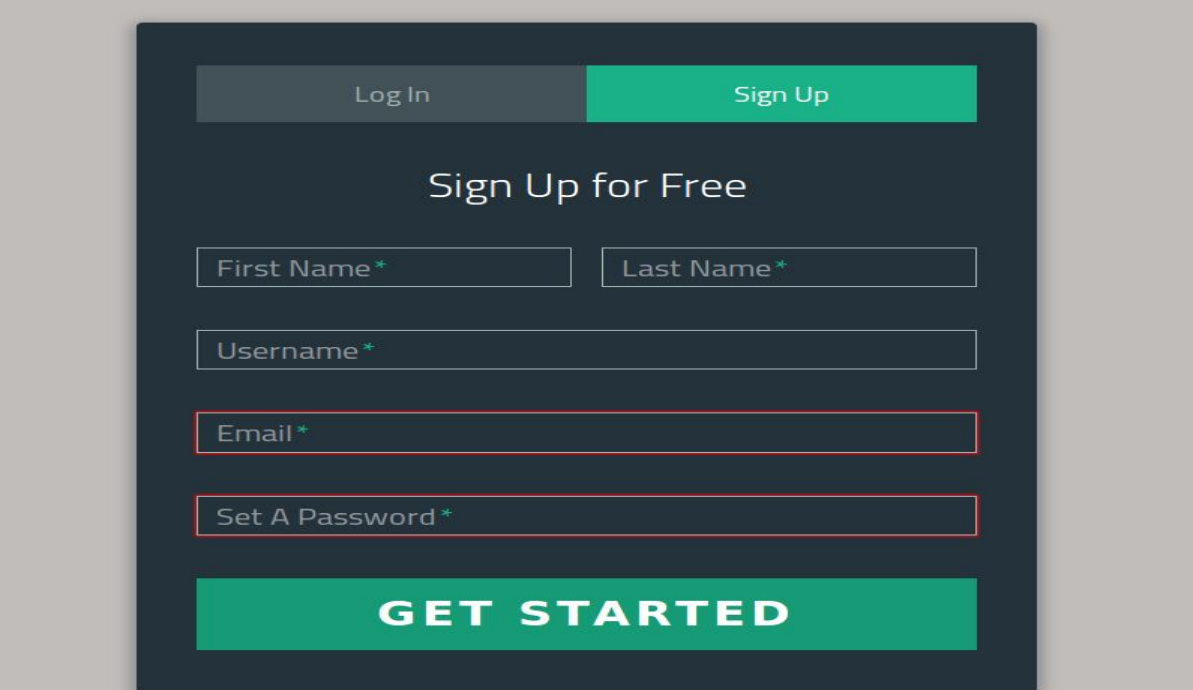
## Réalisation

Pour accéder à l'application, il faut tout d'abord se connecter avec un nom d'utilisateur et un mot de passe.



A login form with a dark blue background. At the top, there are two buttons: 'Log In' (green) and 'Sign Up' (grey). Below them is the text 'Welcome Back!'. There are two input fields: 'Username\*' and 'Password\*'. To the right of the password field is a link 'Forgot Password?'. At the bottom is a large green button with the text 'LOG IN'.

ou bien s'inscrire par le formulaire d'inscription :

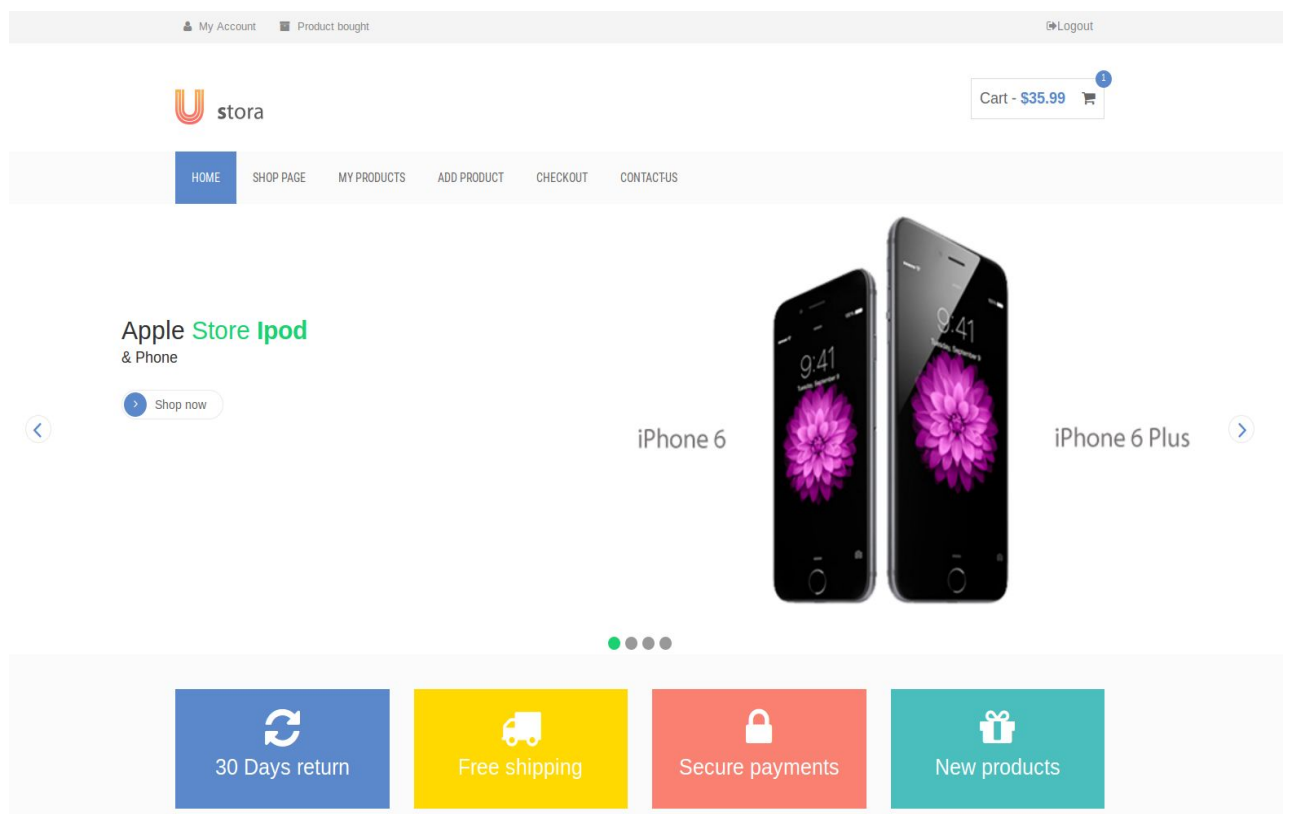


A sign up form with a dark blue background. At the top, there are two buttons: 'Log In' (grey) and 'Sign Up' (green). Below them is the text 'Sign Up for Free'. There are five input fields: 'First Name\*', 'Last Name\*', 'Username\*', 'Email\*', and 'Set A Password\*'. At the bottom is a large green button with the text 'GET STARTED'.

## 1- Page d'accueil

Après l'identification, l'utilisateur se dirige vers sa page d'accueil . Cette dernière contient des informations sur l'app et un menu principal contenant six sous menus : Home, Shop page, My Products, add Product, Checkout, Contact-us.

L'utilisateur peut modifier son compte au bien se déconnecter par les liens au plus haut de la page (My account & logout), ou bien accéder à tout moment au panier (bouton Cart ) et voir le prix global et nombre de produit sur ce dernier.



il peut aussi voir les dernier produits ajoutés

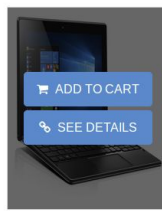
## Latest Products



Kettler Laufrad Speedy  
\$35.99



Benq 9H. Leelb QBC 27 inch  
\$189.00



Lenovo Miix 310 10.1  
\$199.00



Fire, 17.7 cm (7") Display  
\$59.99



Echo Portable  
\$60.99

prev next

NOKIA

Canon

SAMSUNG

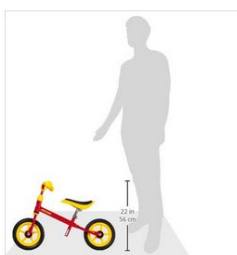
APPLE

## 2- Shop page

Dans cette page l'utilisateur peut voir tous les produits sur l'application ou bien les produits par catégorie en sélectionnant une catégorie sur le menu en haut. Les produits se chargent chaque fois par AJAX pour éviter le chargement long de la page en cas de plusieurs produits.

Multimedia Books & Audible Sports & Outdoors Automotive & Industrial Toys, Kids & Baby Clothing, Shoes & Jewelry Electronics & Computers

## Products



Kettler Laufrad Speedy  
\$ 35.99



Benq 9H. Leelb QBC 27 inch  
\$ 189.00



Lenovo Miix 310 10.1  
\$ 199.00



Fire, 17.7 cm (7") Display  
\$ 59.99



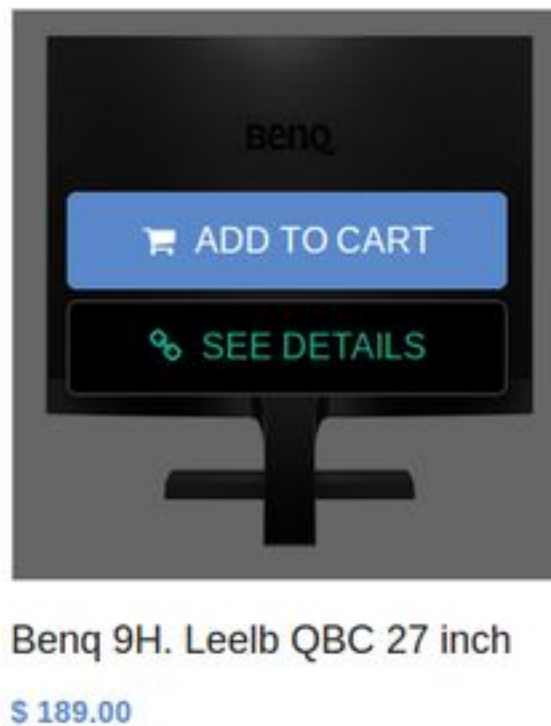
Echo Portable



PC HP i5

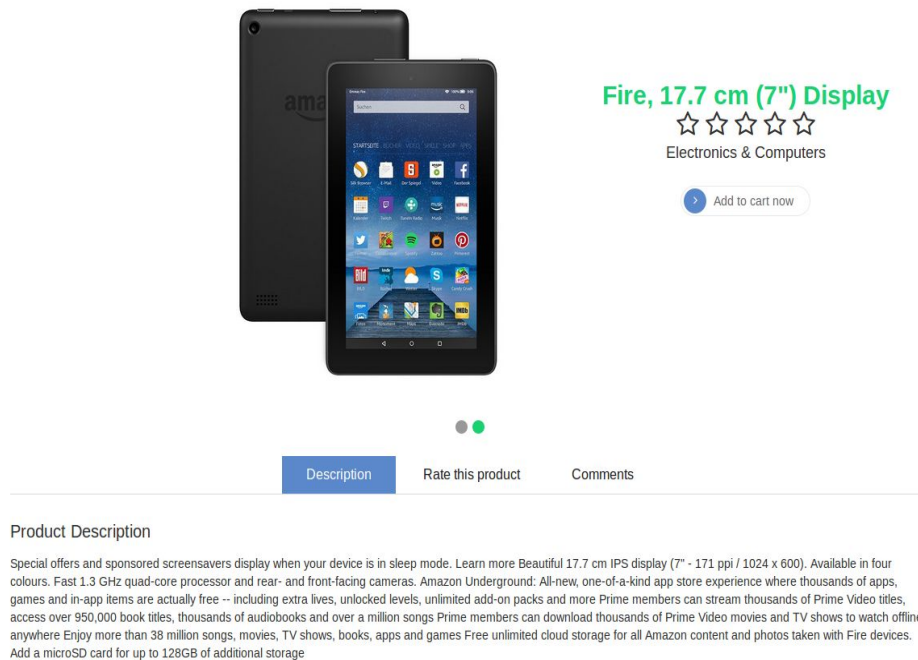
à tout moment l'utilisateur peut rechercher un produit ou ajouter le produit au panier sont rechargement de la page et qui va recalculer le prix total et le nombre de produit directement par AJAX.

l'utilisateur peut aussi voir les détail d'un produit en appuyant sur le bouton 'detail'



### 3- Détail d'un produit

Le détail du produit contient tous les informations sur le produits

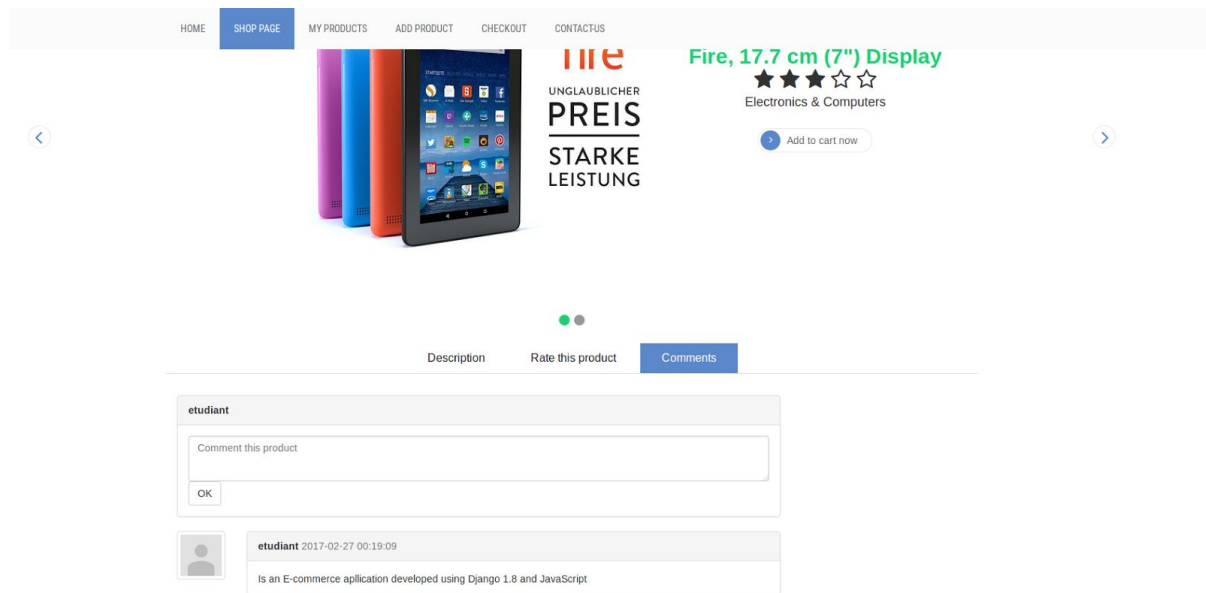


Amazon Fire tablet product page. The main image shows a black Fire tablet with its screen on, displaying the Amazon app store interface. To the right, the text reads: "Fire, 17.7 cm (7") Display", followed by five stars and "Electronics & Computers". Below this is a button that says "Add to cart now". At the bottom, there are three tabs: "Description", "Rate this product", and "Comments". The "Description" tab is active.

**Product Description**

Special offers and sponsored screensavers display when your device is in sleep mode. Learn more Beautiful 17.7 cm IPS display (7" - 171 ppi / 1024 x 600). Available in four colours. Fast 1.3 GHz quad-core processor and rear- and front-facing cameras. Amazon Underground: All-new, one-of-a-kind app store experience where thousands of apps, games and in-app items are actually free -- including extra lives, unlocked levels, unlimited add-on packs and more Prime members can stream thousands of Prime Video titles, access over 950,000 book titles, thousands of audiobooks and over a million songs Prime members can download thousands of Prime Video movies and TV shows to watch offline anywhere Enjoy more than 38 million songs, movies, TV shows, books, apps and games Free unlimited cloud storage for all Amazon content and photos taken with Fire devices. Add a microSD card for up to 128GB of additional storage

l'utilisateur peut noter ou bien commenter ce dernier



E-commerce application interface. The top navigation bar includes links: HOME, SHOP PAGE, MY PRODUCTS, ADD PRODUCT, CHECKOUT, and CONTACTUS. The main content area features a carousel of Fire tablets. The first slide shows a black tablet with the text "Fire, 17.7 cm (7") Display", five stars, "Electronics & Computers", and a button "Add to cart now". The second slide shows a red tablet with the text "UNGLAUBLICHER PREIS" and "STARKE LEISTUNG". Below the carousel are three tabs: "Description", "Rate this product", and "Comments". The "Comments" tab is active.

**etudiant**

Comment this product

OK

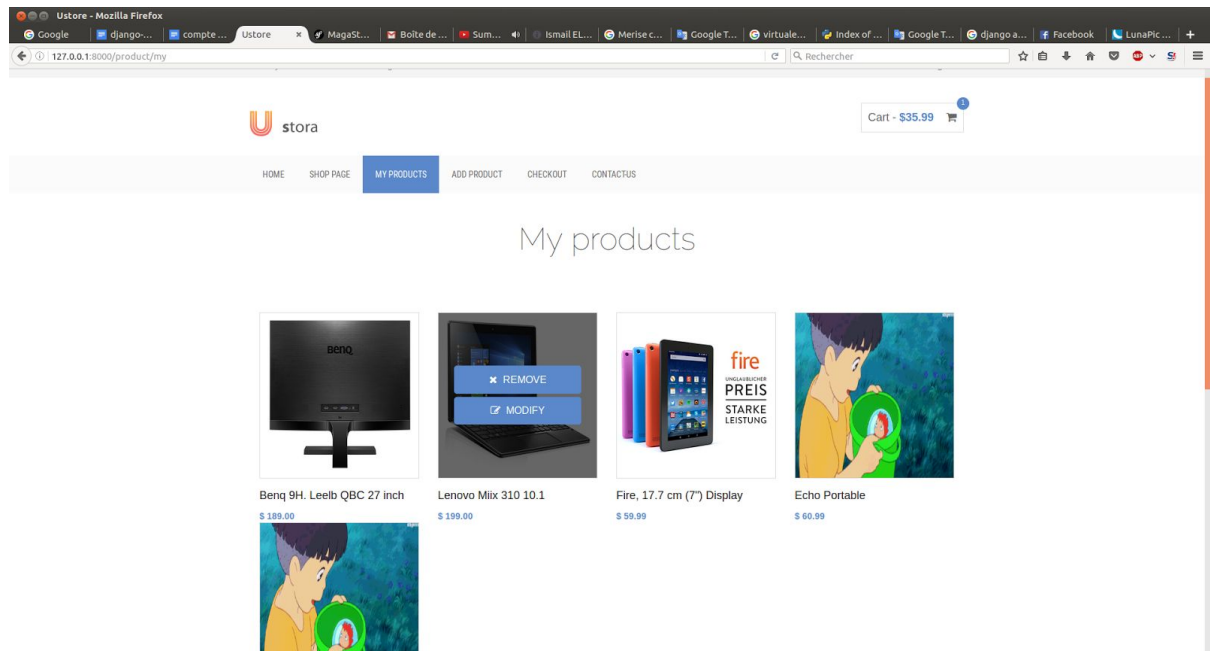
**etudiant** 2017-02-27 00:19:09

Is an E-commerce application developed using Django 1.8 and JavaScript

## 4- Gestion des produites

L'utilisateur peut modifier ou supprimer ses produites





ou bien ajouter un produit en appuyant sur le sous-menu “Add product” et en remplissant le formulaire d’ajout

My Account Product bought Logout

Ustore Cart - \$35.99

HOME SHOP PAGE MY PRODUCTS ADD PRODUCT CHECKOUT CONTACTUS

Name:

Price

Categorie

Description of this product

Veuillez compléter ce champ.

Images (The last one selected will be the official one)


Parcourir... Aucun fichier sélectionné.

ADD

## 5- Panier



L'utilisateur peut voir la liste des produits ajoutés au panier et toutes les informations de ces derniers, la quantité pour chaque produit, et le prix total.

Il peut aussi supprimer un produit du panier, et modifier la quantité. Cela va envoyer des requêtes AJAX sans chargement de la page pour effectuer les modifications et recalculer tous les prix.



Cart - \$35.99

HOME SHOP PAGE MY PRODUCTS ADD PRODUCT CHECKOUT CONTACTUS

		PRODUCT	PRICE	QUANTITY	TOTAL
		Kettler Laufrad Speedy	\$35.99	1	\$35.99


Coupon:  Apply Coupon

CART TOTALS

Shipping and Handling	Free Shipping
Order Total	\$35.99




Buy

En appuyant sur le bouton Buy (acheter) tous les produits seront considérés comme achetés et l'utilisateur peut les voir sur le sous-menu CHECKOUT ou bien en haut 'Product bought'



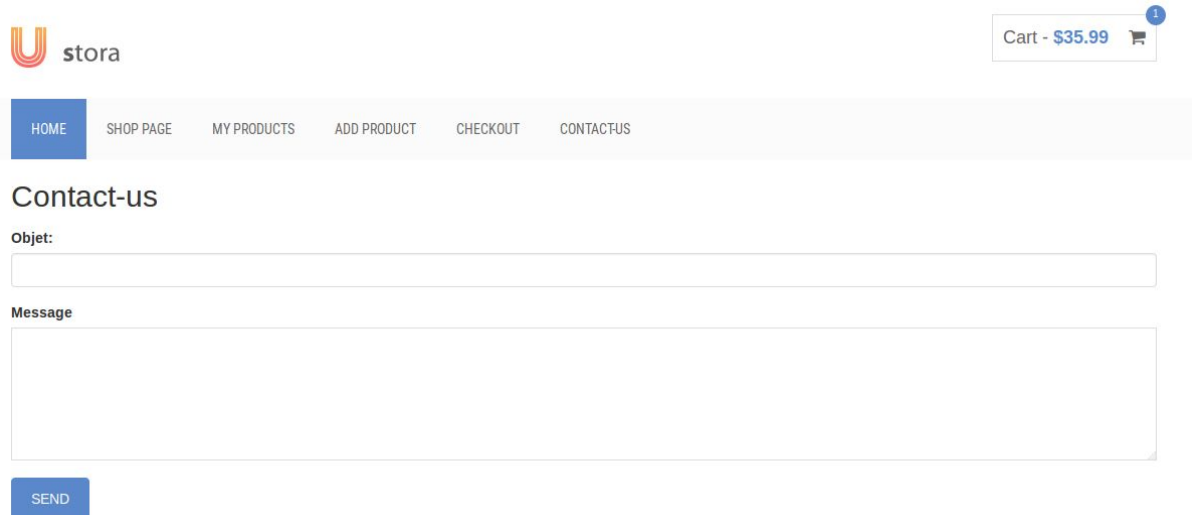
Cart - \$35.99

HOME SHOP PAGE MY PRODUCTS ADD PRODUCT CHECKOUT CONTACTUS

	PRODUCT	PRICE	QUANTITY	TOTAL
	PC HP i5	\$500.24	2	\$1000.48
	Echo Portable	\$60.99	1	\$60.99
	Benq 9H. Leelb QBC 27 inch	\$189.00	2	\$378.00

## 6- Contactez-nous

Les utilisateurs peuvent contactez l'administrateur de l'application en remplissant le formulaire "contact-us" qui permet d'envoyer un email au administrateur qui à comme login "admin"



## 7- Administration

Pour que l'administrateur puisse modifier tous les table sur la base de données j'ai ajouté sur (admin.py) les lignes suivantes :

```
admin.site.register(Categorie)
admin.site.register(Product)
admin.site.register(Cart)
admin.site.register(RatingProduct)
admin.site.register(CommentProduct)
```

Et pour créer un super administrateur j'ai exécuter la commande suivante :

**python manage.py createsuperuser**

aver username = admin, password= minda, et email= [elfaqir963@gmail.com](mailto:elfaqir963@gmail.com)

Site administration

Authentication and Authorization	
Groups	<a href="#">Add</a> <a href="#">Change</a>
Users	<a href="#">Add</a> <a href="#">Change</a>
Client	
Carts	<a href="#">Add</a> <a href="#">Change</a>
Categories	<a href="#">Add</a> <a href="#">Change</a>
Comment products	<a href="#">Add</a> <a href="#">Change</a>
Products	<a href="#">Add</a> <a href="#">Change</a>
Rating products	<a href="#">Add</a> <a href="#">Change</a>

Recent Actions
<b>My Actions</b>
<a href="#">etudiant</a> Cart
<a href="#">etudiant</a> Cart
<a href="#">etudiant</a> Cart
<a href="#">etudiant</a> User
<a href="#">Product Fire, 17.7 cm (7") Display</a> Product
<a href="#">Product Benq 9H. Leelb QBC 27 Inch</a> Product
<a href="#">Echo Portable</a> Comment product
<a href="#">client</a> Group
<a href="#">client</a> Group
<a href="#">Product PC HP i5</a> Product