Міністерство освіти і науки України

НТУУ «Київський політехнічний інститут

ім. Ігоря Сікорського»

Фізико-технічний інститут

Курсова робота

з курсу програмування для Web

тема : «Інтернет-магазин»

**Виконав:**

Студент 2 курсу ФТІ

групи ФІ-61

Хоменко Р.О.

**Київ 2017**

Зміст

[Формулювання вимог 4](#_Toc500712554)

[Опис прецедентів 5](#_Toc500712555)

[Діаграма прецедентів 6](#_Toc500712556)

(діагр.) [6](#_Toc500712557)

[Модель баз даних(діаграма) 7](#_Toc500712558)

[Модель предметної області 7](#_Toc500712559)

[Модель проектування 9](#_Toc500712560)

[Засоби розробки 9](#_Toc500712561)

[Вимоги до обладнання 9](#_Toc500712562)

[Код проекту 10](#_Toc500712563)

[Шаблони 17](#_Toc500712564)

# Формулювання вимог

У даній курсовій роботі потрібно було реалізувати зручний інтерфейс перегляду та

замовлення товарів, корзину(можливість додання/видалення товарів),систему

швидкої реєстрації користувача.

# Опис прецедентів

Користувач(потенційний покупець)— людина, яка заходить на сайт з метою придбання товарів.

Основнi сценарiї:

1. Перегляд наявних товарів
2. Додання товарів в корзину
3. Видалення товарів з корзини
4. Придбання товарів
5. Пошук товарів
6. Авторизація

Адмiнiстратор — людина, що має доступ до всiх функцiї адмiнiстрування

системи. Адмiнiстратор може переглядати,додавати,створювати, редагувати, видаляти товари.Також адміністратор може керувати користувачами.

Основні сценарії:

1. Додати товар
2. Видалити товар
3. Створити користувача
4. Видалити користувача

Список прецедентів:

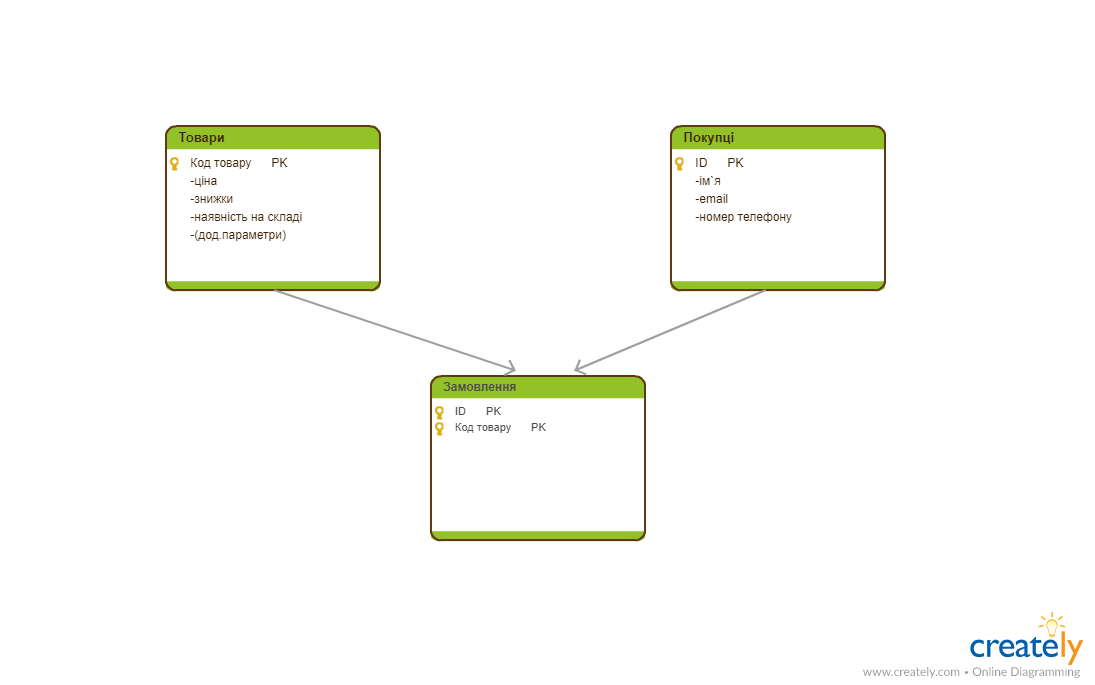
Користувач заходить на сайт,бачить перелік всіх наявних товарів на даний момент.Користувач може додати товари,які йому сподобалися,до корзини.Перейшовши до корзини,Користувач може видалити непотрібні товари,а також побачити загальну вартість товарів,які знаходяться в корзині. Якщо потрібні товари обрані,загальна сума задовольняє Користувача,то Користувач може перейти до реєстрації:заповнює відповідну форму,вводить свої контактні дані.Дані про Користувача,а також придбані ним товари передаються до БД.

Адміністратор має доступ до БД,до всіх користувачів і придбаних ними товарів.Адміністратор може створювати чи видаляти користувачів.Адміністратор має доступ до всіх товарів тому може видаляти наявні на сайті товари,чи додавати нові товари з відповідними параметрами(ціна,опис,фото і т.д.).

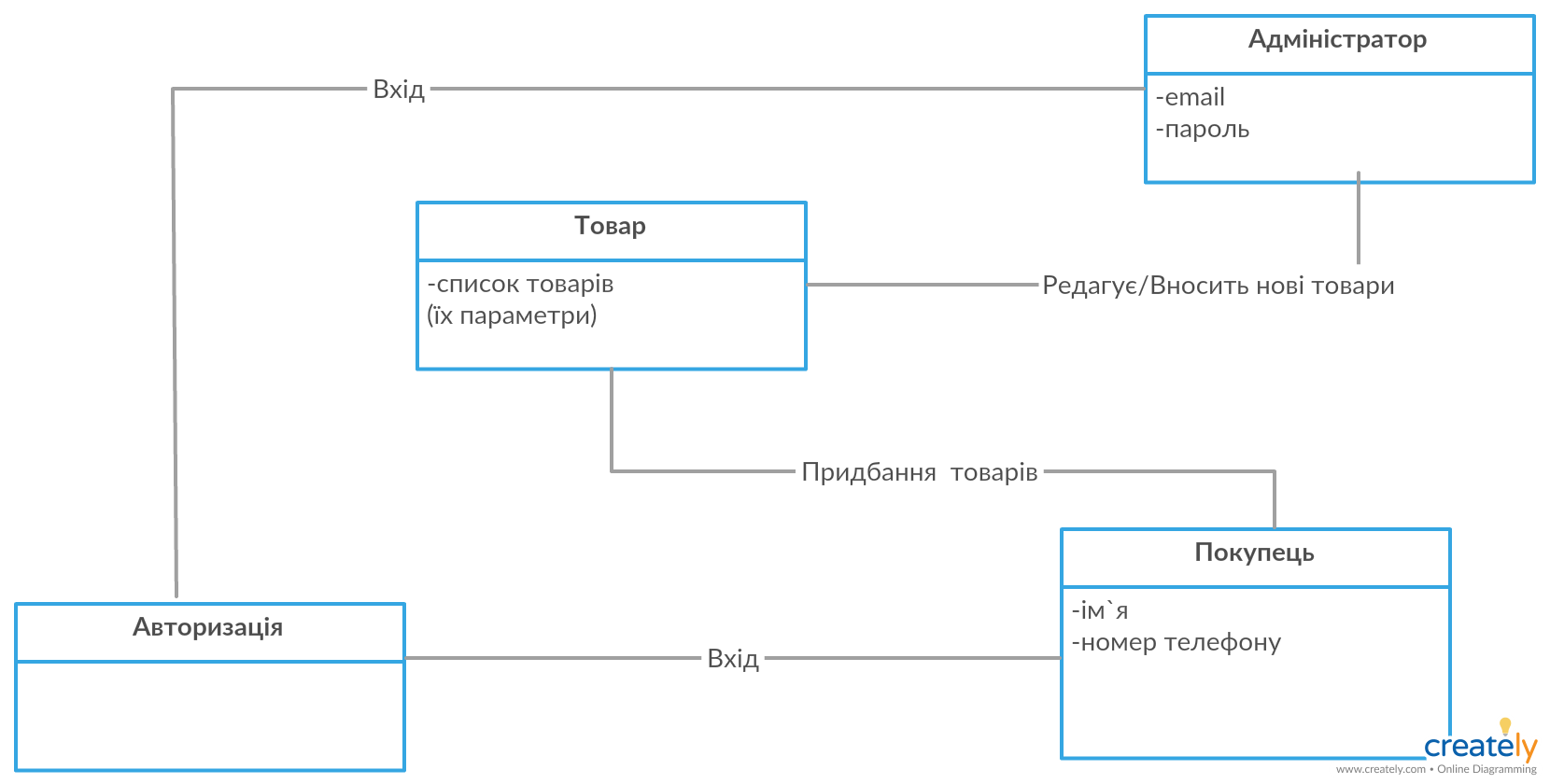
# Діаграма прецедентів

# C:\Users\Asus\Desktop\New Online Shopping System erer (5).png

### Модель баз даних(діаграма)

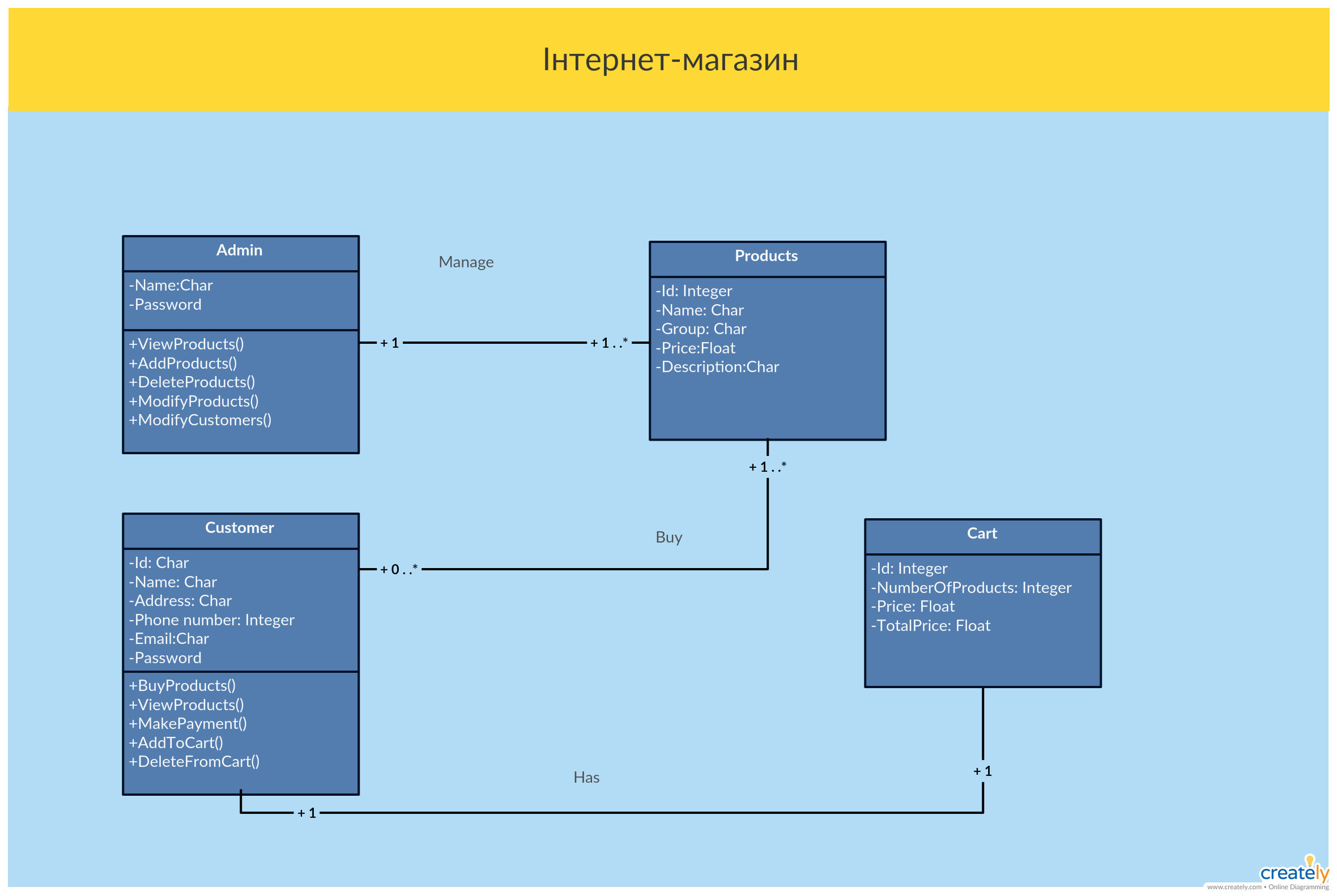


### Модель предметної області





### Модель проектування



# Засоби розробки

Побудова сайту відбувалась за допомогою фреймворку Django.

Для роботи з базою даних необхiдно встановити пакет sqlite.Однак при встановленні Python ми використали версію з уже вбудованим пакетом SQlite.Пiд час роботи був використаний браузер Google Chrome. Був вибраний саме цей браузер, бо вiн має ряд переваг, якi зробили комфортнiшою роботу з сайтом. Iншими використаними готовими засобами є HTML, CSS, JavaScript, JQuery,технологія AJAX.

# Вимоги до обладнання

Робота була виконана на апаратному засобi з наступною конфiгурацiєю:

1. Оперативна пам’ять: 4 Гб
2. Процесори: 4
3. Жорсткий диск: 824 Гб
4. Операцiйна система: Windows 10

# Код проекту

ORDERS

Admin.py

**from** django.contrib **import** admin  
**from** .models **import** \*  
  
  
**class** ProductInOrderInline(admin.TabularInline):  
 model = ProductInOrder  
 extra = 0  
  
  
**class** StatusAdmin (admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** Status.\_meta.fields]  
  
 **class** Meta:  
 model = Status  
  
admin.site.register(Status, StatusAdmin)  
  
  
**class** OrderAdmin (admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** Order.\_meta.fields]  
 inlines = [ProductInOrderInline]  
  
 **class** Meta:  
 model = Order  
  
admin.site.register(Order, OrderAdmin)  
  
  
**class** ProductInOrderAdmin (admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** ProductInOrder.\_meta.fields]  
  
 **class** Meta:  
 model = ProductInOrder  
  
admin.site.register(ProductInOrder, ProductInOrderAdmin)

context\_processors.py

**from** .models **import** ProductInBasket  
  
**def** getting\_basket\_info(request):  
 session\_key = request.session.session\_key  
 **if not** session\_key:  
 request.session.cycle\_key()  
  
 products\_in\_basket = ProductInBasket.objects.filter(session\_key=session\_key, is\_active=True,order\_\_isnull=True)  
 products\_total\_numberofprod = products\_in\_basket.count()  
  
 **return** locals()

models.py

**from** django.db **import** models  
**from** products.models **import** Product  
**from** django.db.models.signals **import** post\_save  
**from** django.contrib.auth.models **import** User  
  
  
**class** Status(models.Model):  
 name = models.CharField(max\_length=24, blank=True, null=True, default=None)  
 is\_active = models.BooleanField(default=True)  
 created = models.DateTimeField(auto\_now\_add=True, auto\_now=False)  
 updated = models.DateTimeField(auto\_now\_add=False, auto\_now=True)  
  
 **def** \_\_str\_\_(self):  
 **return "Статус %s"** % self.name  
  
 **class** Meta:  
 verbose\_name = **'Статус замовлення'** verbose\_name\_plural = **'Статуси замолення'  
  
  
class** Order(models.Model):  
 user = models.ForeignKey(User, blank=True, null=True, default=None)  
 total\_price = models.DecimalField(max\_digits=20, decimal\_places=2, default=0)*#total price for all products in order* customer\_name = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 customer\_phone = models.CharField(max\_length=48, blank=True, null=True, default=None)  
 customer\_address = models.CharField(max\_length=128, blank=True, null=True, default=None)  
 status = models.ForeignKey(Status)  
 created = models.DateTimeField(auto\_now\_add=True, auto\_now=False)  
 updated = models.DateTimeField(auto\_now\_add=False, auto\_now=True)  
  
 **def** \_\_str\_\_(self):  
 **return "Замовлення %s %s"** % (self.id, self.status.name)  
  
 **class** Meta:  
 verbose\_name = **'Замовлення'** verbose\_name\_plural = **'Замовлення'  
  
 def** save(self, \*args, \*\*kwargs):  
  
 super(Order, self).save(\*args, \*\*kwargs)  
  
  
**class** ProductInOrder(models.Model):  
 order = models.ForeignKey(Order, blank=True, null=True, default=None)  
 numberofprod=models.IntegerField(default=1)  
 product = models.ForeignKey(Product, blank=True, null=True, default=None)  
 price\_per\_item = models.DecimalField(max\_digits=20, decimal\_places=2, default=0)  
 total\_price = models.DecimalField(max\_digits=20, decimal\_places=2, default=0)*#price\*numberofprod* is\_active = models.BooleanField(default=True)  
 created = models.DateTimeField(auto\_now\_add=True, auto\_now=False)  
 updated = models.DateTimeField(auto\_now\_add=False, auto\_now=True)  
  
 **def** \_\_str\_\_(self):  
 **return "%s"** % self.product.name  
  
 **class** Meta:  
 verbose\_name = **'Товар в замовленні'** verbose\_name\_plural = **'Товари в замовленні'  
  
  
 def** save(self, \*args, \*\*kwargs):  
 price\_per\_item = self.product.price  
 self.price\_per\_item = price\_per\_item  
 self.total\_price = int(self.numberofprod) \* price\_per\_item  
  
 super(ProductInOrder, self).save(\*args, \*\*kwargs)  
  
  
  
**def** product\_in\_order\_post\_save(sender, instance, created, \*\*kwargs):  
 order = instance.order  
 all\_products\_in\_order = ProductInOrder.objects.filter(order=order, is\_active=True)  
  
 order\_total\_price = 0  
 **for** item **in** all\_products\_in\_order:  
 order\_total\_price += item.total\_price  
  
 instance.order.total\_price = order\_total\_price  
 instance.order.save(force\_update=True)  
  
  
post\_save.connect(product\_in\_order\_post\_save, sender=ProductInOrder)  
  
  
**class** ProductInBasket(models.Model):  
 session\_key = models.CharField(max\_length=128, blank=True, null=True, default=None)  
 order = models.ForeignKey(Order, blank=True, null=True, default=None)  
 product = models.ForeignKey(Product, blank=True, null=True, default=None)  
 numberofprod = models.IntegerField(default=1)  
 price\_per\_item = models.DecimalField(max\_digits=20, decimal\_places=2, default=0)  
 total\_price = models.DecimalField(max\_digits=20, decimal\_places=2, default=0)*#price\*numberofprod* is\_active = models.BooleanField(default=True)  
 created = models.DateTimeField(auto\_now\_add=True, auto\_now=False)  
 updated = models.DateTimeField(auto\_now\_add=False, auto\_now=True)  
  
 **def** \_\_str\_\_(self):  
 **return "%s"** % self.product.name  
  
 **class** Meta:  
 verbose\_name = **'Товар в корзині'** verbose\_name\_plural = **'Товари в корзині'  
  
 def** save(self, \*args, \*\*kwargs):  
 price\_per\_item = self.product.price  
 self.price\_per\_item = price\_per\_item  
 self.total\_price = price\_per\_item \* int(self.numberofprod)  
  
 super(ProductInBasket, self).save(\*args, \*\*kwargs)

urls.py

**from** django.conf.urls **import** url, include  
**from** django.contrib **import** admin  
**from** orders **import** views  
  
urlpatterns = [  
 url(**r'^basket\_adding/$'**, views.basket\_adding, name=**'basket\_adding'**),  
 url(**r'^checkout/$'**, views.checkout, name=**'checkout'**),  
]

views.py

**from** .models **import** ProductInBasket  
**from** django.shortcuts **import** render  
**from** django.http **import** JsonResponse, HttpResponse, HttpResponseRedirect  
**from** .models **import** \*  
**from** forms **import** CheckoutContactForm  
**from** django.contrib.auth.models **import** User  
  
  
**def** basket\_adding(request):  
 return\_dict = dict()  
 session\_key = request.session.session\_key  
 **print** (request.POST)  
 data = request.POST  
 product\_id = data.get(**"product\_id"**)  
 numberofprod = data.get(**"numberofprod"**)  
 is\_delete = data.get(**"is\_delete"**)  
  
 **if** is\_delete == **'true'**:  
 ProductInBasket.objects.filter(id=product\_id).update(is\_active=False)  
 **else**:  
 new\_product, created = ProductInBasket.objects.get\_or\_create(session\_key=session\_key, product\_id=product\_id,  
 is\_active=True, defaults={**"numberofprod"**: numberofprod})  
 **if not** created:  
 **print** (**"not created"**)  
 new\_product.numberofprod += int(numberofprod)  
 new\_product.save(force\_update=True)  
  
 *#common code for 2 cases* products\_in\_basket = ProductInBasket.objects.filter(session\_key=session\_key, is\_active=True)  
 products\_total\_numberofprod = products\_in\_basket.count()  
 return\_dict[**"products\_total\_numberofprod"**] = products\_total\_numberofprod  
  
 return\_dict[**"products"**] = list()  
  
 **for** item **in** products\_in\_basket:  
 product\_dict = dict()  
 product\_dict[**"id"**] = item.id  
 product\_dict[**"name"**] = item.product.name  
 product\_dict[**"price\_per\_item"**] = item.price\_per\_item  
 product\_dict[**"numberofprod"**] = item.numberofprod  
 return\_dict[**"products"**].append(product\_dict)  
  
 **return** JsonResponse(return\_dict)  
  
  
**def** checkout(request):  
 session\_key = request.session.session\_key  
 products\_in\_basket = ProductInBasket.objects.filter(session\_key=session\_key, is\_active=True, order\_\_isnull=True)  
 **print** (products\_in\_basket)  
 **for** item **in** products\_in\_basket:  
 **print**(item.order)  
  
  
 form = CheckoutContactForm(request.POST **or** None)  
 **if** request.POST:  
 **print**(request.POST)  
 **if** form.is\_valid():  
 **print**(**"yes"**)  
 data = request.POST  
 name = data.get(**"name"**, **"3423453"**)  
 phone = data[**"phone"**]  
 user, created = User.objects.get\_or\_create(username=phone, defaults={**"first\_name"**: name})  
  
 order = Order.objects.create(user=user, customer\_name=name, customer\_phone=phone, status\_id=2)  
  
 **for** name, value **in** data.items():  
 **if** name.startswith(**"product\_in\_basket\_"**):  
 product\_in\_basket\_id = name.split(**"product\_in\_basket\_"**)[1]  
 product\_in\_basket = ProductInBasket.objects.get(id=product\_in\_basket\_id)  
 **print**(type(value))  
  
 product\_in\_basket.numberofprod = value  
 product\_in\_basket.order = order  
 product\_in\_basket.save(force\_update=True)  
  
 ProductInOrder.objects.create(product=product\_in\_basket.product, numberofprod = product\_in\_basket.numberofprod,  
 price\_per\_item=product\_in\_basket.price\_per\_item,  
 total\_price = product\_in\_basket.total\_price,  
 order=order)  
  
 **return** HttpResponseRedirect(request.META[**'HTTP\_REFERER'**])  
 **else**:  
 **print**(**"no"**)  
 **return** render(request, **'orders/checkout.html'**, locals())

PRODUCTS

Models.py

**from** django.db **import** models  
  
**class** ProductCategory(models.Model):  
 name = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 is\_active = models.BooleanField(default=True)  
  
 **def** \_\_str\_\_(self):  
 **return "%s"** % self.name  
  
 **class** Meta:  
 verbose\_name = **'Категорія товару'** verbose\_name\_plural = **'Категорія товарів'  
  
  
  
  
class** Product(models.Model):  
 name = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 price = models.DecimalField(max\_digits=20, decimal\_places=2, default=0)  
 discount = models.IntegerField(default=0)  
 category = models.ForeignKey(ProductCategory, blank=True, null=True, default=None)  
 short\_description = models.TextField(blank=True, null=True, default=None)  
 description = models.TextField(blank=True, null=True, default=None)  
 is\_active = models.BooleanField(default=True)  
 created = models.DateTimeField(auto\_now\_add=True, auto\_now=False)  
 updated = models.DateTimeField(auto\_now\_add=False, auto\_now=True)  
  
 **def** \_\_str\_\_(self):  
 **return "%s, %s"** % (self.price, self.name)  
  
 **class** Meta:  
 verbose\_name = **'Товар'** verbose\_name\_plural = **'Товари'  
  
  
class** ProductImage(models.Model):  
 product = models.ForeignKey(Product, blank=True, null=True, default=None)  
 image = models.ImageField(upload\_to=**'products\_images/'**)  
 is\_main = models.BooleanField(default=False)  
 is\_active = models.BooleanField(default=True)  
 created = models.DateTimeField(auto\_now\_add=True, auto\_now=False)  
 updated = models.DateTimeField(auto\_now\_add=False, auto\_now=True)  
  
 **def** \_\_str\_\_(self):  
 **return "%s"** % self.id  
  
 **class** Meta:  
 verbose\_name = **'Фотографія'** verbose\_name\_plural = **'Фотографії'**

urls.py

**from** django.conf.urls **import** url, include  
**from** django.contrib **import** admin  
**from** products **import** views  
  
urlpatterns = [  
 *#url(r'^start/$', views.start, name='start'),* url(**r'^product/(?P<product\_id>\w+)/$'**, views.product, name=**'product'**),  
]

views.py

**from** django.shortcuts **import** render  
**from** products.models **import** \*  
  
  
  
**def** product(request, product\_id):  
 product = Product.objects.get(id=product\_id)  
 session\_key = request.session.session\_key  
 **if not** session\_key:  
 request.session.cycle\_key()  
  
 **print**(request.session.session\_key)  
  
 **return** render(request, **'products/product.html'**, locals())

START

urls.py

**from** django.conf.urls **import** url, include  
**from** django.contrib **import** admin  
**from** start **import** views  
  
urlpatterns = [  
 url(**r'^$'**, views.home, name=**'home'**),  
]

views.py

**from** django.shortcuts **import** render  
**from** products.models **import** \*  
  
  
**def** home(request):  
 products\_images = ProductImage.objects.filter(is\_active=True, is\_main=True, product\_\_is\_active=True)  
 products\_images\_mens = products\_images.filter(product\_\_category\_\_id=1)  
 products\_images\_womans = products\_images.filter(product\_\_category\_\_id=2)  
 **return** render(request, **'start/home.html'**, locals())

scripts.js

**$**(**document**).ready(**function**(){  
 **var** form = **$**(**'#form\_buying\_product'**);  
 **console**.log(form);  
  
  
 **function** *basketUpdating*(product\_id, numberofprod, is\_delete){  
 **var** data = {};  
 data.**product\_id** = product\_id;  
 data.**numberofprod** = numberofprod;  
 **var** csrf\_token = **$**(**'#form\_buying\_product [name="csrfmiddlewaretoken"]'**).val();  
 data[**"csrfmiddlewaretoken"**] = csrf\_token;  
  
 **if** (is\_delete){  
 data[**"is\_delete"**] = **true**;  
 }  
  
 **var** url = form.attr(**"action"**);  
  
 **console**.log(data)  
 **$**.**ajax**({  
 **url**: url,  
 **type**: **'POST'**,  
 **data**: data,  
 **cache**: **true**,  
 success: **function** (data) {  
 **console**.log(**"OK"**);  
 **console**.log(data.products\_total\_numberofprod);  
 **if** (data.products\_total\_numberofprod || data.products\_total\_numberofprod == 0){  
 **$**(**'#basket\_total\_numberofprod'**).text(**"("**+data.products\_total\_numberofprod+**")"**);  
 **console**.log(data.products);  
 **$**(**'.basket-items ul'**).**html**(**""**);  
 **$**.each(data.products, **function**(k, v){  
 **$**(**'.basket-items ul'**).append(**'<li>'**+ v.**name**+**', '** + v.**numberofprod** + **'шт. '** + **'по '** + v.price\_per\_item + **'грн '** +  
 **'<a class="delete-item" href="" data-product\_id="'**+v.**id**+**'">x</a>'**+  
 **'</li>'**);  
 });  
 }  
  
 },  
 error: **function**(){  
 **console**.log(**"error"**)  
 }  
 })  
  
 }  
  
 form.on(**'submit'**, **function**(e){  
 e.preventDefault();  
 **console**.log(**'123'**);  
 **var** numberofprod = **$**(**'#number'**).val();  
 **console**.log(numberofprod);  
 **var** submit\_btn = **$**(**'#submit\_btn'**);  
 **var** product\_id = submit\_btn.data(**"product\_id"**);  
 **var** name = submit\_btn.data(**"name"**);  
 **var** price = submit\_btn.data(**"price"**);  
 **console**.log(product\_id );  
 **console**.log(name);  
  
 *basketUpdating*(product\_id, numberofprod, **is\_delete**=**false**)  
  
 });  
  
 **function** *showingBasket*(){  
 **$**(**'.basket-items'**).removeClass(**'hidden'**);  
 };  
**$**(**'.basket-container'**).mouseover(**function**(){  
 *showingBasket*();  
  
 });  
  
**$**(**document**).on(**'click'**, **'.delete-item'**, **function**(e){  
 e.preventDefault();  
 **product\_id** = **$**(**this**).data(**"product\_id"**)  
 **numberofprod** = 0;  
 *basketUpdating*(**product\_id**, **numberofprod**, **is\_delete**=**true**)  
 })  
  
  
  
 **function** *calculatingBasketAmount*(){  
 **var** total\_order\_amount = 0;  
 **$**(**'.total-product-in-basket-amount'**).each(**function**() {  
 total\_order\_amount = total\_order\_amount + parseFloat(**$**(**this**).text());  
 });  
 **console**.log(total\_order\_amount);  
 **$**(**'#total\_order\_amount'**).text(total\_order\_amount.toFixed(2));  
 };  
  
 **$**(**document**).on(**'change'**, **".product-in-basket-numberofprod"**, **function**(){  
 **var** current\_numberofprod = **$**(**this**).val();  
 **console**.log(current\_numberofprod);  
  
 **var** current\_tr = **$**(**this**).closest(**'tr'**);  
 **var** current\_price = parseFloat(current\_tr.**find**(**'.product-price'**).text()).toFixed(2);  
 **console**.log(current\_price);  
 **var** total\_amount = parseFloat(current\_numberofprod\*current\_price).toFixed(2);  
 **console**.log(total\_amount);  
 current\_tr.**find**(**'.total-product-in-basket-amount'**).text(total\_amount);  
  
 *calculatingBasketAmount*();  
 });  
  
  
 *calculatingBasketAmount*();  
  
});

# Шаблони

Checkout.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
  
{% **block content** %}  
 <**div class="container"**>  
 <**div class="row"**>  
 {% **if products\_in\_basket** %}  
 <**div class="col-lg-8 col-lg-offset-2"**>  
 <**h3 class="text-center"**>Товари в корзині</**h3**>  
*{# <h5 class="text-center">Загальна вартість: {{ request.session.basket.total\_amount }} грн </h5>#}  
{# {{ request.session.basket.products }}#}* <**form class="form-horizontal" method="post"**>{% **csrf\_token** %}  
  
 <**div class="form-group"**>  
 <**div class="col-lg-4"**>  
 <**label**>Ім'я</**label**>  
 <**span class="form-error"**>{{ **form**.**name**.**errors** }}</**span**>  
 <**input placeholder="" type="text" name="**{{ **form**.**name**.**html\_name** }}**" class="form-control"  
 value="**{{ **request**.**POST**.**name** }}**"** >  
 </**div**>  
 </**div**>  
  
 <**div class="form-group"**>  
 <**div class="col-lg-4"**>  
 <**label**>Телефон</**label**>  
 <**span class="form-error"**>{{ **form**.**phone**.**errors** }}</**span**>  
 <**input type="tel" placeholder="(999) 999-99-99" pattern="\([0-9]{3}\)\s[0-9]{3}-[0-9]{2}-[0-9]{2}" name="**{{ **form**.**phone**.**html\_name** }}**" class="form-control"  
 value="**{{ **request**.**POST**.**phone** }}**"** >  
 </**div**>  
 </**div**>  
  
  
 <**table class="table text-center"**>  
 <**thead**>  
 <**tr**>  
 <**th**>Назва</**th**>  
 <**th**>Кількість</**th**>  
 <**th**>Ціна за одиницю</**th**>  
 <**th**>Загальна ціна</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 {% **for product\_in\_basket in products\_in\_basket** %}  
 <**tr**>  
 <**td**>  
 {{ **product\_in\_basket**.**product**.**name** }}  
 </**td**>  
 <**td**>  
 <**input value="**{{ **product\_in\_basket**.**numberofprod** }}**" type="number"  
 class="product-in-basket-numberofprod" name="product\_in\_basket\_**{{ **product\_in\_basket**.**id** }}**"**>  
 </**td**>  
 <**td**>  
 <**span class="product-price"**>  
 {{ **product\_in\_basket**.**price\_per\_item**}}  
 </**span**>  
 </**td**>  
 <**td**>  
 <**span class="total-product-in-basket-amount"**>  
 {{ **product\_in\_basket**.**total\_price** }}  
 </**span**>  
 </**td**>  
 </**tr**>  
 {% **endfor** %}  
 </**tbody**>  
 </**table**>  
  
 <**div class="text-right"**>  
 <**b**>Сума замовлення:</**b**> <**span id="total\_order\_amount"**></**span**>  
 </**div**>  
  
 <**hr**>  
  
 <**div class="text-right"**>  
 <**button type="submit" class="btn btn-primary btn-lg"**>Замовити</**button**>  
 </**div**>  
  
 </**form**>  
  
 </**div**>  
 {% **else** %}  
 <**h3 class="text-center"**>Ваша корзина порожня</**h3**>  
 {% **endif** %}  
  
 </**div**>  
 </**div**>  
{% **endblock** %}

Home.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
  
{% **block content** %}  
 <**div class="section-top"**>  
 <**img src="**{% **static 'img/1nb.jpg'** %}**" class="img-responsive"**>  
 </**div**>  
 <**div class="section section1"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-lg-12"**>  
 <**h1 class="text-center"**>  
 Нові надходження  
 </**h1**>  
 </**div**>  
 {% **for product\_image in products\_images** %}  
 {% **include 'start/product\_item.html'** %}  
  
 {% **endfor** %}  
  
  
 </**div**>  
 </**div**>  
  
 </**div**>  
  
 <**div class="section section2 mens-container"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-lg-12"**>  
 <**h1 class="text-center"**>  
 Чоловічі кросівки  
 </**h1**>  
 </**div**>  
 {% **for product\_image in products\_images\_mens** %}  
 {% **include 'start/product\_item.html'** %}  
 {% **endfor** %}  
  
  
 </**div**>  
 </**div**>  
  
  
 </**div**>  
 <**div class="section section3 womans-container"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-lg-12"**>  
 <**h1 class="text-center"**>  
 Жіночі кросівки  
 </**h1**>  
 </**div**>  
 {% **for product\_image in products\_images\_womans** %}  
 {% **include 'start/product\_item.html'** %}  
 {% **endfor** %}  
  
 </**div**>  
 </**div**>  
 </**div**>  
  
<**script src="**{% **static 'js/scripts.js'** %}**"**></**script**>  
  
{% **endblock** %}

Product\_item.html

<**div class="col-lg-3"**>  
 <**div class="product-item"**>  
 <**div**>  
 <**a href="**{% **url 'product' product\_image**.**product**.**id** %}**"**>  
 <**img src="**{{ **product\_image**.**image**.**url** }}**" class="img-responsive"**>  
 </**a**>  
 </**div**>  
  
 {% **if product\_image**.**product**.**discount** %}  
 <**div class="discount-container"**>  
 <**span**>{{ **product\_image**.**product**.**discount** }}%</**span**>  
 </**div**>  
 {% **endif** %}  
  
 <**h4**>{{ **product\_image**.**product**.**name** }}</**h4**>  
 <**p**>  
 {{ **product\_image**.**product**.**description**|**truncatechars\_html**:70 }}  
 </**p**>  
 <**div**>  
 {{ **product\_image**.**product**.**price** }} UAH  
 </**div**>  
 <**div class="add-to-card-btn"**>  
 <**button class="btn btn-warning"**>  
 <**a id="btndb" href="**{% **url 'product' product\_image**.**product**.**id** %}**"**>Переглянути</**a**>  
 </**button**>  
 </**div**>  
 </**div**>  
</**div**>