**Содержание**

[Цели 3](#_Toc505034930)

[Задачи 3](#_Toc505034931)

[Введение 4](#_Toc505034932)

[Техническое задание 5](#_Toc505034933)

[Основание для разработки 5](#_Toc505034934)

[Назначение разработки 5](#_Toc505034935)

[Требования к программе 5](#_Toc505034936)

[Требования к функциональным характеристикам 5](#_Toc505034937)

[Требования к составу и параметрам технических средств 6](#_Toc505034938)

[Требования к информационной и программной совместимости 7](#_Toc505034939)

[Стадии и этапы разработки 7](#_Toc505034940)

[Анализ и проектирование базы данных предметной области 8](#_Toc505034941)

[Анализ разделов сайта и выделение логических разделов сайта 15](#_Toc505034942)

[Тестирование программы 18](#_Toc505034943)

[Заключение 29](#_Toc505034944)

[Приложение 30](#_Toc505034945)

[Приложение 1. Исходный код 30](#_Toc505034946)

[Settings.py 30](#_Toc505034947)

[Urls.py 32](#_Toc505034948)

[wsgi.py 32](#_Toc505034949)

[orders/admin.py 32](#_Toc505034950)

[orders/apps.py 34](#_Toc505034951)

[orders/context\_processors.py 34](#_Toc505034952)

[orders/forms.py 34](#_Toc505034953)

[orders/models.py 34](#_Toc505034954)

[orders/urls.py 37](#_Toc505034955)

[orders/views.py 37](#_Toc505034956)

[products/admin.py 38](#_Toc505034957)

[products/apps.py 39](#_Toc505034958)

[products/forms.py 39](#_Toc505034959)

[products/models.py 39](#_Toc505034960)

[products/urls.py 40](#_Toc505034961)

[products/views.py 40](#_Toc505034962)

[static/static\_dev/css/style.css 40](#_Toc505034963)

[static/static\_dev/js/scripts.js 45](#_Toc505034964)

[templates/landing/home.html 46](#_Toc505034965)

[templates/login\_auth/login.html 48](#_Toc505034966)

[templates/login\_auth/logout.html 48](#_Toc505034967)

[templates/login\_auth/register.html 48](#_Toc505034968)

[templates/login\_auth/view\_profile.html 48](#_Toc505034969)

[templates/products/product.html 49](#_Toc505034970)

[templates/products/show\_products.html 50](#_Toc505034971)

[templates/publications/base.html 50](#_Toc505034972)

[templates/publications/footer.html 51](#_Toc505034973)

[templates/publications/header.html 51](#_Toc505034974)

[templates/publications/navbar.html 52](#_Toc505034975)

[Список использованных источников 55](#_Toc505034976)

Цели

Разработать WEB - приложение, которое предоставит пользователю полную информацию об интересующей продукции и возможность заказать товар с выбором способа оплаты и доставки.

Задачи

Для реализации функционала приложения необходимо разработать:

Пользовательский интерфейс

Алгоритм регистрации пользователя в системе

Алгоритм создания и оформления заказа

Административный модуль управления сайтом

Базу данных

Введение

В настоящее время Интернет становиться все более развитой средой для осуществления коммуникаций с потребителями. В тоже время, существенным является и тот факт, что Интернет становиться удобной и достаточно дешевой «торговой площадкой». Все большее количество фирм старается представить свою продукцию в on-line среде. При этом такое представление не ограничивается только лишь созданием промо-сайтов и размещением рекламных баннеров и статей в электронных журналах и на информационных порталах. С развитием Интернет-среды развивается и само предложение. Теперь люди могут не только получать интересующую их информацию, но и совершать покупки.

Студия гончарных изделий так же нуждается в собственной платформе для продажи своих изделий в сети Интернет. Заказчиком сайта является мастер, согласно требованиям которого будет осуществляться работа над проектом.

Наименование программы: «Интернет-магазин гончарных изделий Centrobejnaya»

Техническое задание

Основание для разработки

Учебный план по дисциплине «Технология программирования в среде Python», направление бакалавриата 09.03.02 «Информационные системы и технологии», Московский Технологический Университет, 2017 г.

Наименование – «Web-приложение для оформления, демонстрации товаров компании».

Условное обозначение – «Интернет-магазин гончарных изделий Centrobejnaya»

Назначение разработки

Создание WEB - системы, предназначенной для каталогизации гончарных изделий, обеспечения функции интернет-магазина.

Требования к программе

Требования к функциональным характеристикам

Система должна обеспечивать возможность выполнения следующих функций:

• Отображение страниц:

1. Главная
2. Страница товара
3. Страница категории товаров
4. Страница заказа
5. Страница авторизации
6. Страница регистрации
7. Страница профиля
8. Страница редактирования профиля
9. Страница деавторизации

• Отображение информационных блоков: «контакты», «корзина»

• Инициализацию системы (ввод списков покупателей, перечней товаров и т. п.);

• Запись и хранение информации о покупателях и товарах магазина;

• Создание и редактирование учетной записи покупателя, вход и выход из учетной записи;

• Формирование «корзины» с выбранными покупателем товарами;

• Изменение ассортимента товаров;

• Расчет и перерасчет товара при добавлении его в корзину, удаление товара из корзины;

• Формирование заказа.

Требования к составу и параметрам технических средств

Для использования фреймворка Django версии 1.11 нужна версия Python не ниже 3.4. Для хранения информации нужна СУБД версия SQLite не ниже 2.0.

Web-приложение является кросс-платформенным и не зависит от клиентской операционной системы. Web-приложение поддерживает большинство современных и популярных браузеров (Chrome, Mozilla, Safari, Opera) актуальных версий.

Для размещения Web-приложения нужен Ubuntu Server 16.04, для которой есть данные требования:

● Поддержка версии интерпретатора Python не ниже 3.6.2

● Поддержка web-фреймворка Django версии не ниже 2.0

База данных и автоматизированная система развертываются на одном сервере.

Требования к информационной и программной совместимости

При разработке программного продукта используются:

- среда разработки PyCharm Professional 2017.3;

- система управления базами данных (СУБД) – SQLite3, язык SQL;

- языки программирования – JavaScript, CSS, Python 3.6.x (не ниже 3.6.2);

- библиотеки и инструменты – Django 1.11(Python), jQuery.

- язык разметки – HTML5;

- таблицы стилей – CSS;

- CSS фреймворк - Bootstrap 4.

Интерфейс интернет-магазина должен соответствовать требованиям заказчика.

Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

- Разработка технического задания.

- Написание кода web-приложения.

- Тестирование и сопровождение.

При разработке данного проекта были определены следующие этапы:

- формулировка задач и требований к системе;

- исследование предметной области (анализ данных, процессов и т.д.), проектирование базы данных предметной области;

- анализ разделов сайта и выделение логических разделов сайта;

- разработка клиентских приложений(сайт): разработка backend и frontend части приложения;

- отладка и тестирование программы;

- наполнение базы данных товарами;

- составление сопровождающей документации.

Анализ и проектирование базы данных предметной области

Согласованная с Заказчиком схема базы данных (логическая модель) на рисунке 1:

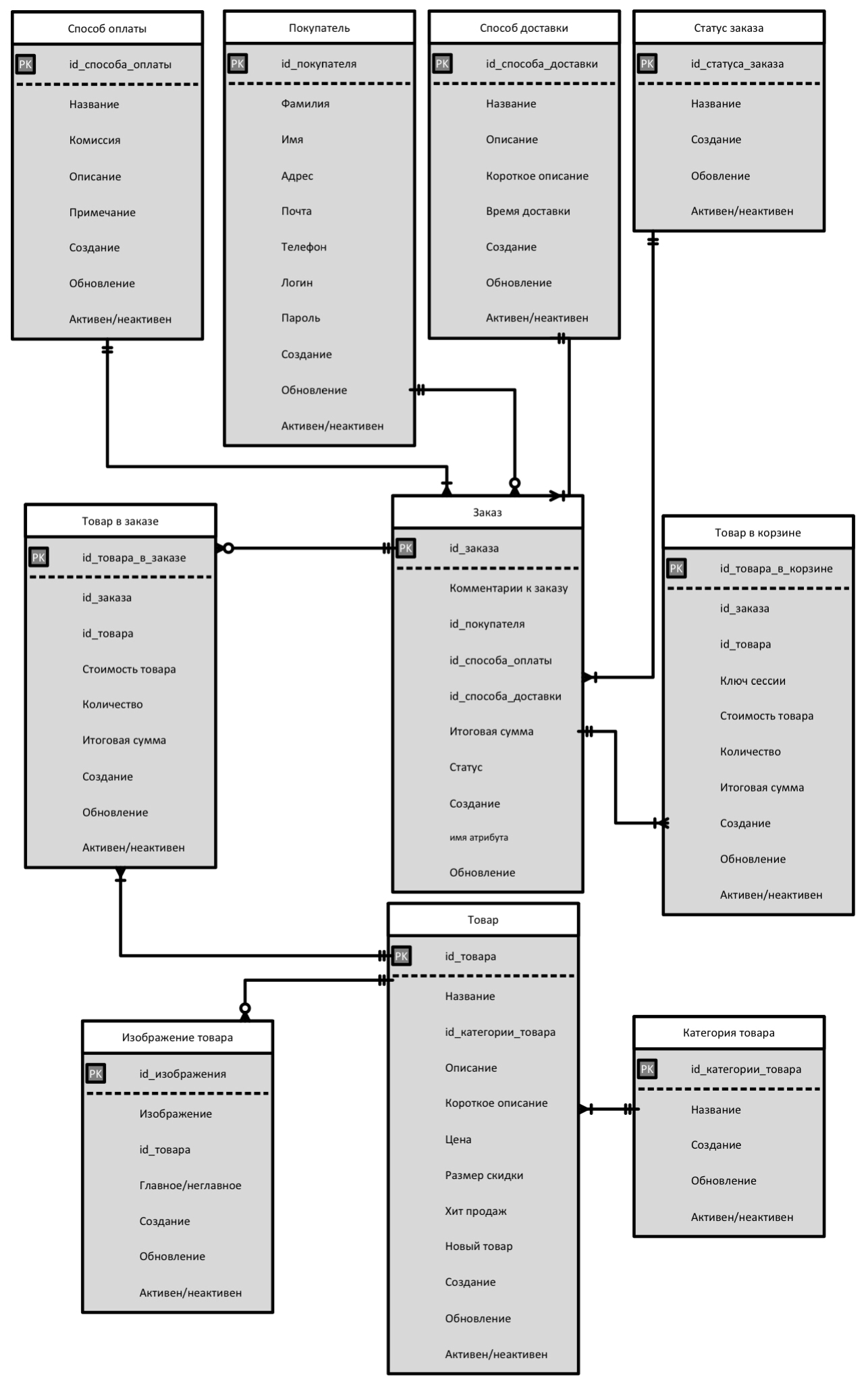


Рисунок 1. Логическая модель базы данных

Физическая модель, рисунок 2:

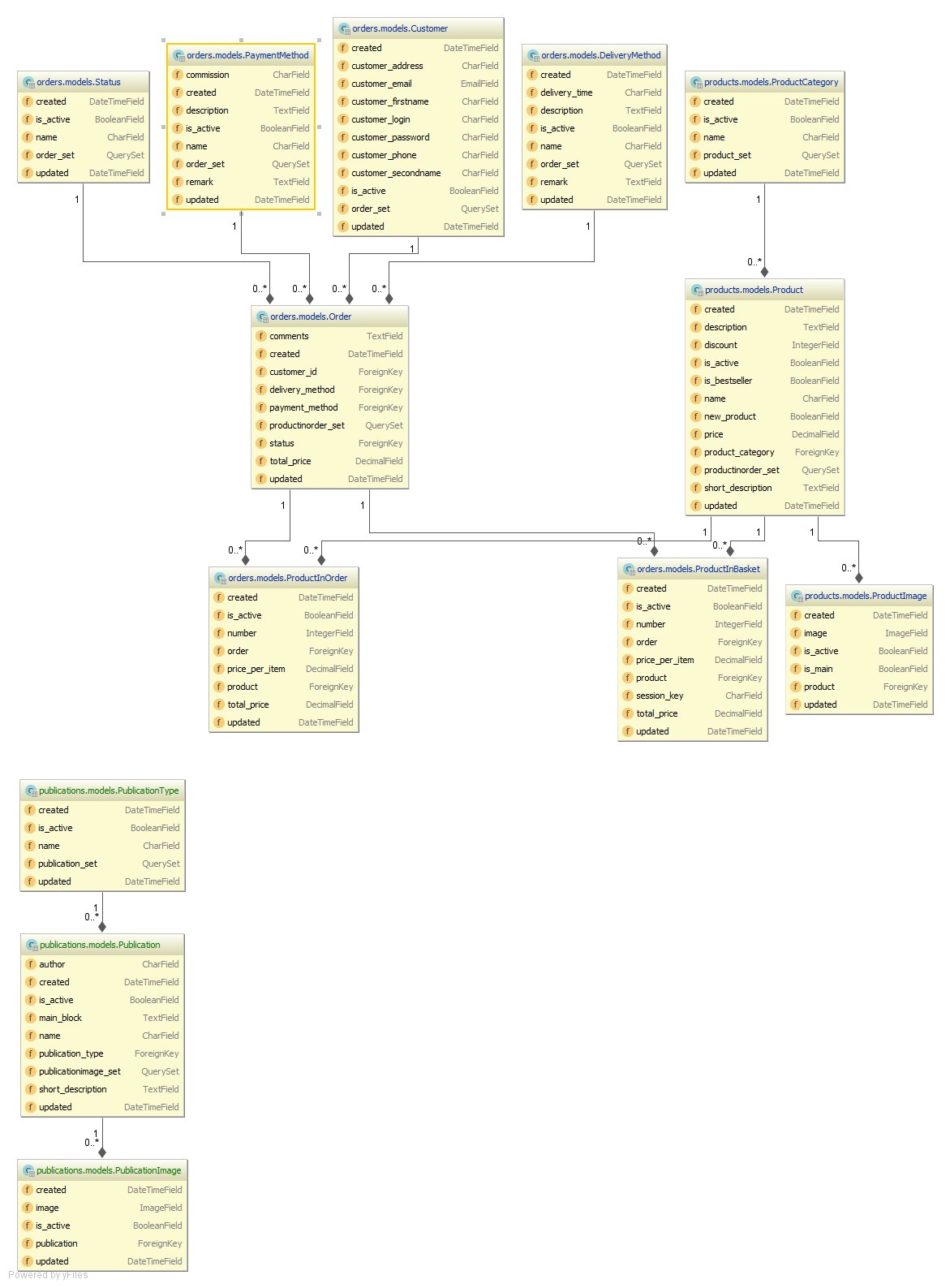


Рисунок 2. Физическая модель базы данных

Выделенные сущности представлены в таблице 1:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Примечание |
| Customer | | |
| customer\_login | Varchar (64) | Логин |
| customer\_password | Varchar (64) | Пароль |
| customer\_firstname | Varchar (64) | Фамилия |
| customer\_secondname | Varchar (64) | Имя |
| customer\_email | Varchar (64) | Почта |
| customer\_phone | Varchar (48) | Телефон |
| customer\_address | Varchar (128) | Адрес |
| is\_active | Boolean | Активен/неактивен |
| Created | Timestamp | Создание |
| Updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| DeliveryMethod | | |
| Name | Varchar (24) | Название |
| Description | Text | Описание |
| delivery\_time | Varchar (24) | Время доставки |
| Remark | Text | Примечание |
| is\_active | Boolean | Активен/неактивен |
| Created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| PaymentMethod | | |
| Name | Varchar (24) | Название |
| Description | Text | Описание |
| commission | Varchar (24) | Комиссия |
| Remark | Text | Примечание |
| is\_active | Boolean | Активен/неактивен |
| Created | Timestamp | Создание |
| Updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| Status | | |
| name | Varchar (24) | Название |
| is\_active | Boolean | Активен/неактивен |
| created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| Order | | |
| total\_price | DecimalField (10) | Итоговая сумма |
| customer\_id | Integer | Primary Key |
| delivery\_method | Integer | Primary Key |
| payment\_method | Integer | Primary Key |
| comments | Text | Комментарий к заказу |
| status | Integer | Primary Key |
| created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| ProductInOrder | | |
| order | Integer | Primary Key |
| product | Integer | Primary Key |
| number | Integer | Количество |
| price\_per\_item | Numeric (10,2) | Стоимость товара |
| total\_price | Numeric (10,2) | Итоговая сумма |
| is\_active | Boolean | Активен/неактивен |
| created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| ProductInBasket | | |
| order | Integer | Primary Key |
| product | Integer | Primary Key |
| number | Integer | Количество |
| price\_per\_item | Numeric (10,2) | Стоимость товара |
| total\_price | Numeric (10,2) | Итоговая сумма |
| session\_key | Varchar (128) | Ключ сессии |
| is\_active | Boolean | Активен/неактивен |
| created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| Product | | |
| name | Varchar (64) | Название |
| price | Numeric (10,2) | Цена |
| discount | Integer | Размер скидки |
| product\_category | Integer | Primary Key |
| short\_description | Text | Короткое описание |
| description | Text | Описание |
| new\_product | Boolean | Новый товар/Не новый товар |
| is\_bestseller | Boolean | Хит продаж/Не хит продаж |
| is\_active | Boolean | Активен/неактивен |
| created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| ProductCategory | | |
| name | Varchar (48) | Название |
| is\_active | Boolean | Активен/неактивен |
| created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

|  |  |  |
| --- | --- | --- |
| ProductImage | | |
| product | Integer | Primary Key |
| image | Varchar (100) | Путь к изображению |
| is\_main | Boolean | Главное изображение/ неглавное |
| is\_active | Boolean | Активен/неактивен |
| created | Timestamp | Создание |
| updated | Timestamp | Редактирование |

Таблица 1

Анализ разделов сайта и выделение логических разделов сайта

Были выделены следующие разделы Web-приложения:

1. Главная
   1. Товары
2. Страница товара
3. Страница категории товаров
4. Страница заказа
5. Страница авторизации
6. Страница регистрации
7. Страница профиля
8. Страница редактирования учетной записи
9. Страница деавторизации

А также информационные блоки: «контакты», «корзина»

Административный раздел:

1. Добавление, удаление, редактирование товаров
2. Добавление, удаление, редактирование категорий
3. Удаление и просмотр заказов

Для frontend составляющей web-приложения использовались инструменты:

* HTML5
* CSS3
* JavaScript
* Bootstrap 4
* AJAX
* jQuery.

Для backend составляющей web-приложения использовался фреймворк Django версии 1.11

Код прикреплен в разделе «Приложение».

Тестирование программы

Главная страница

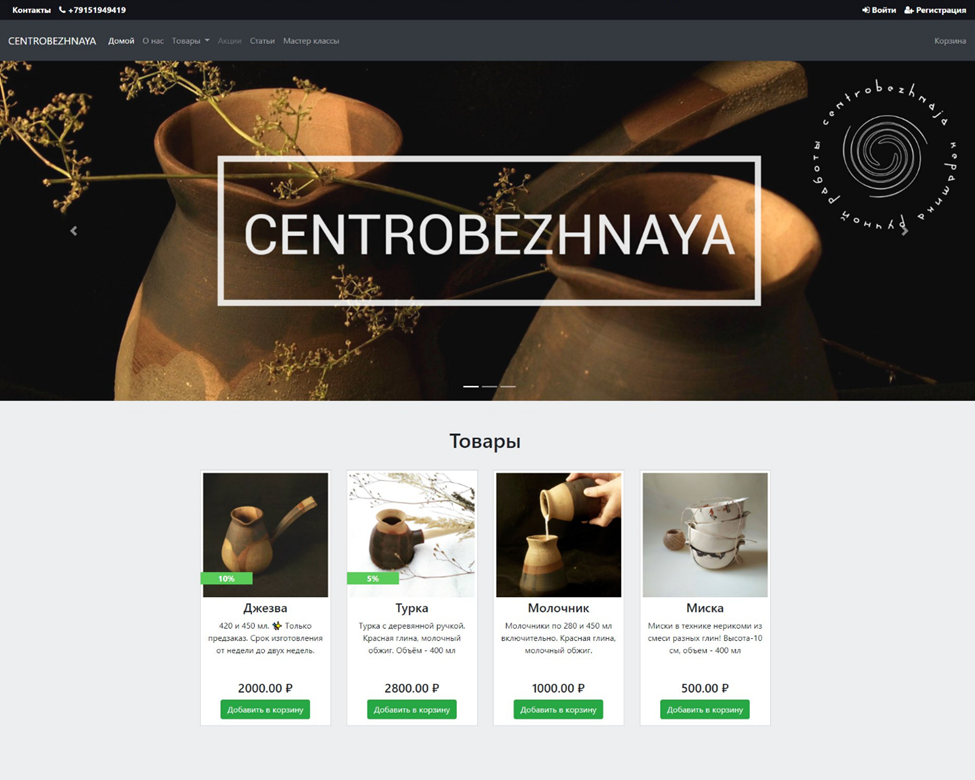


Рисунок 4. Главная страница

Переход с главной страницы на страницу конкретного товара

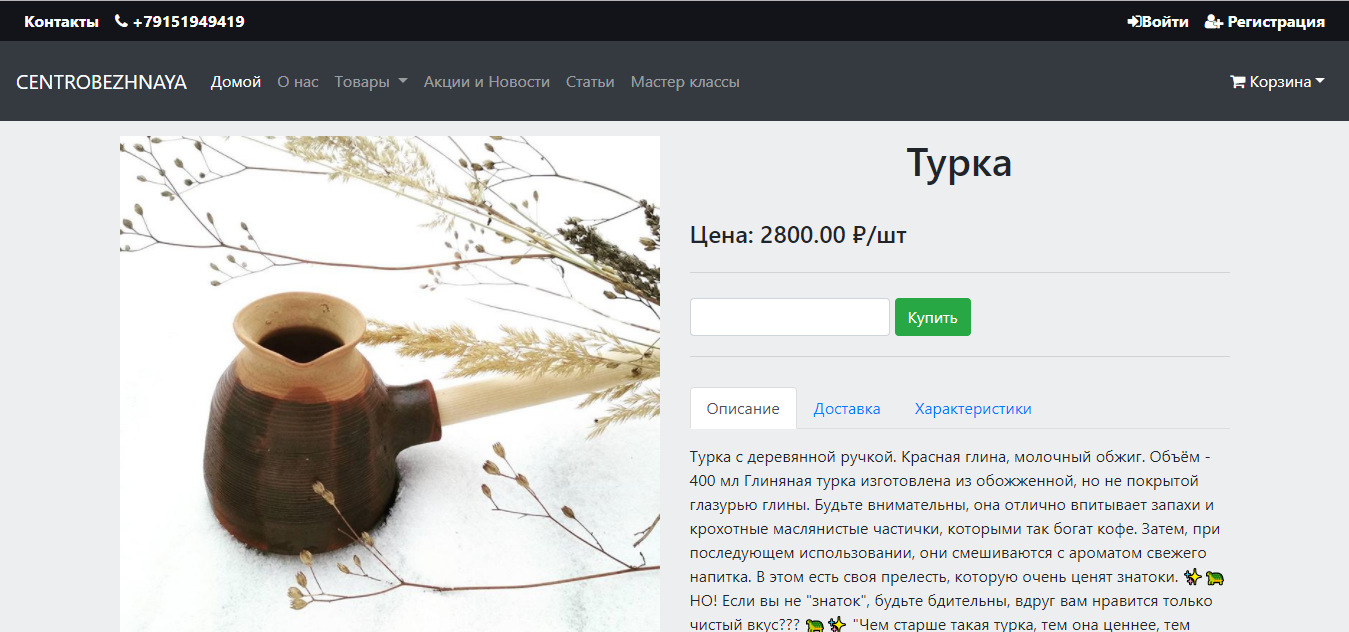


Рисунок 5. Страница товара

Страница регистрации

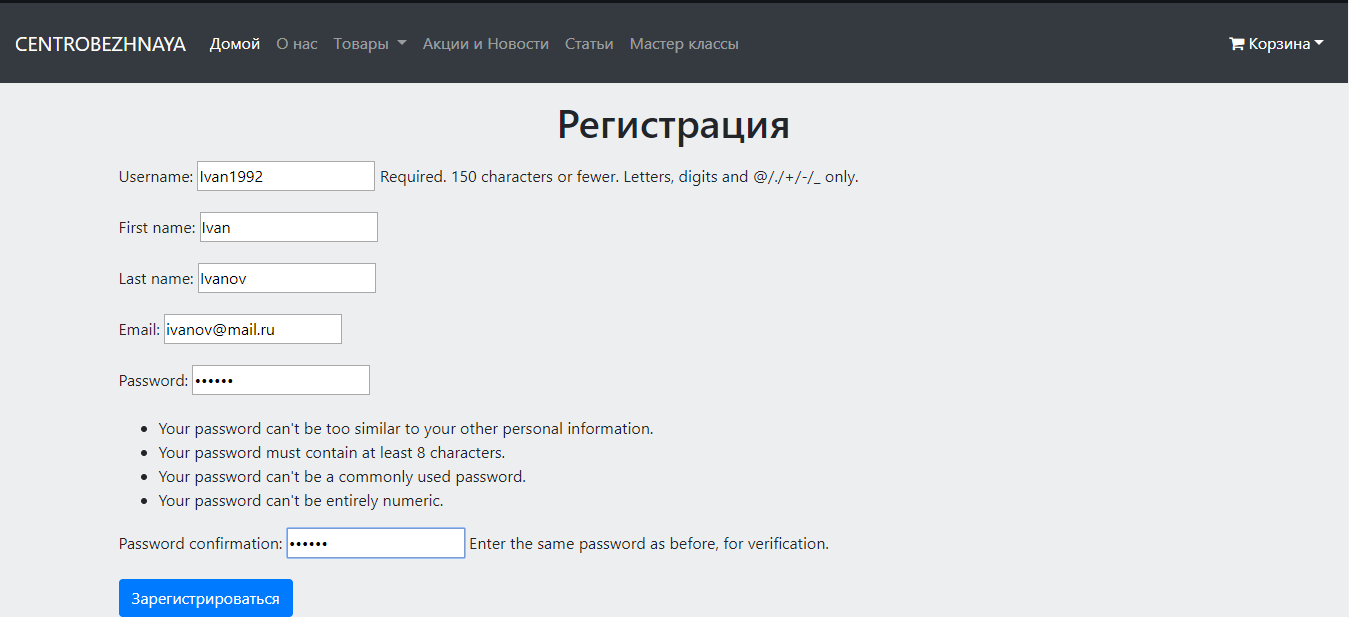


Рисунок 6. Форма регистрации

Вход в учетную запись

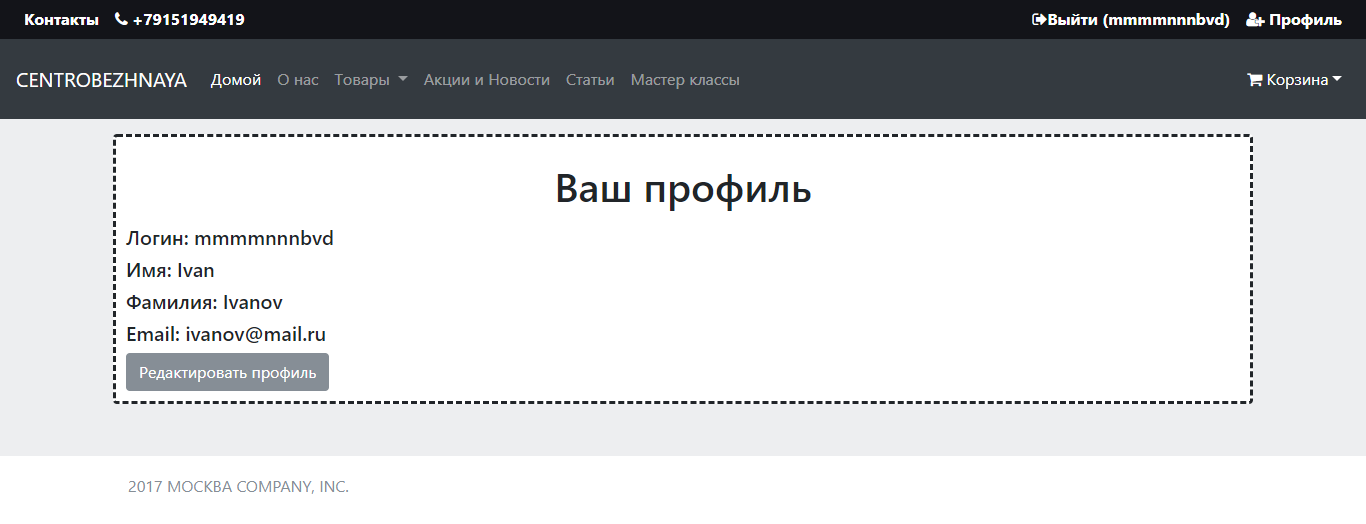


Рисунок 7. Профиль

Редактирование учетной записи

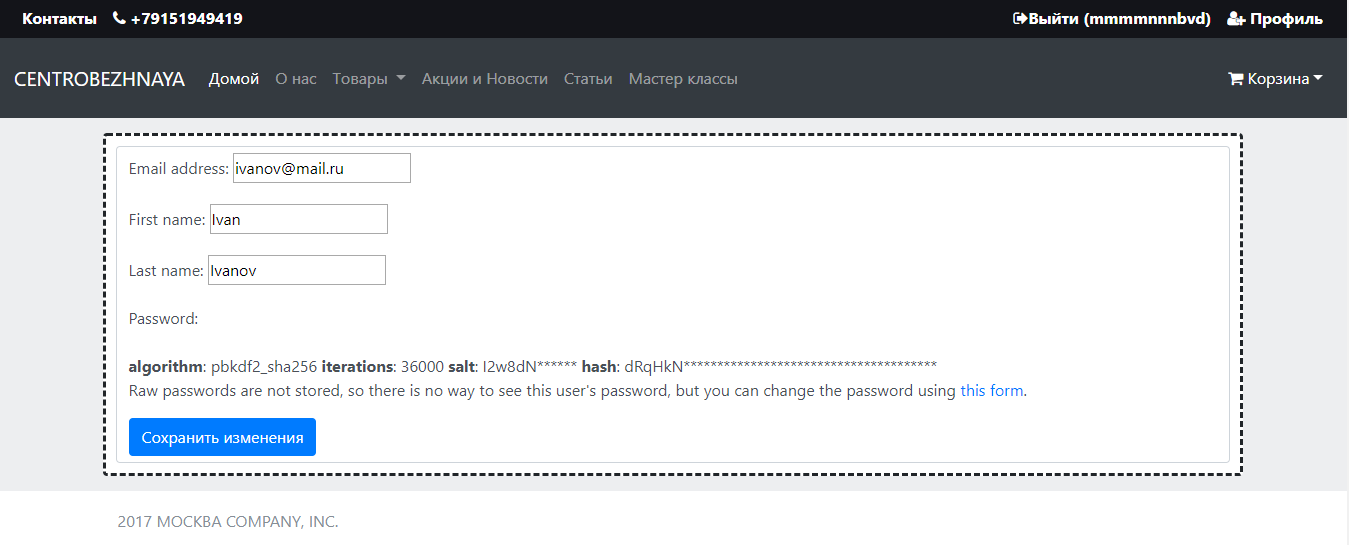


Рисунок 8. Профиль

Информационный блок «контакты»

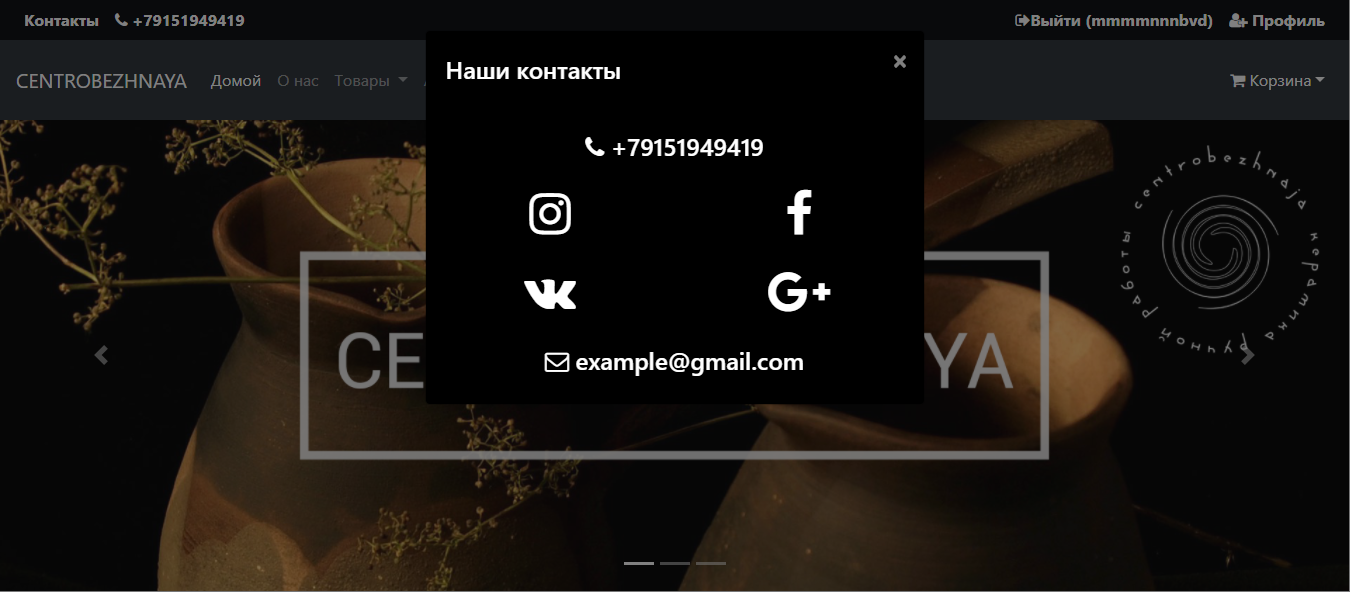


Рисунок 9. Контакты

Отображение корзины

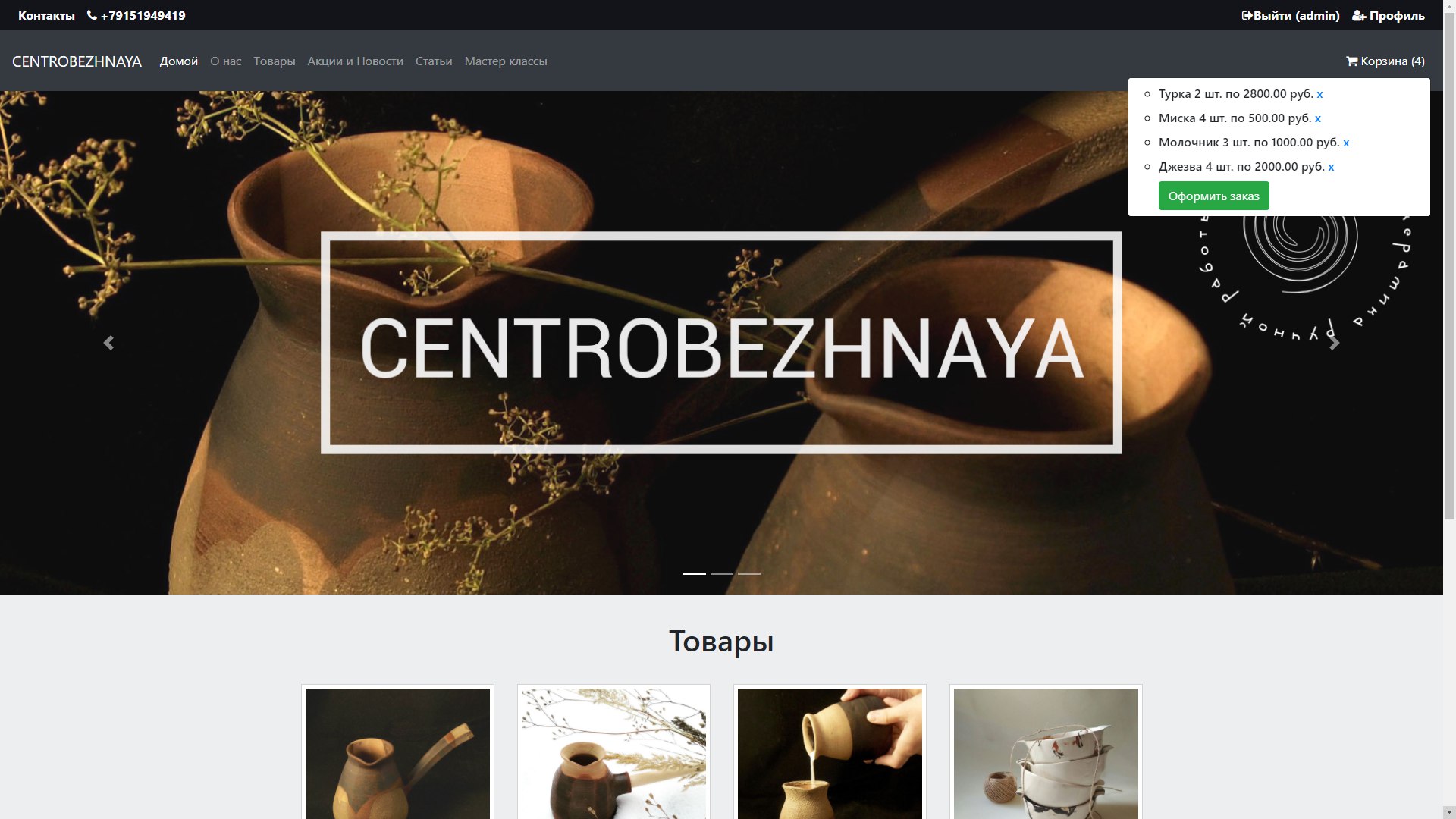


Рисунок 10. Корзина

Переход по кнопке «оформить заказ»

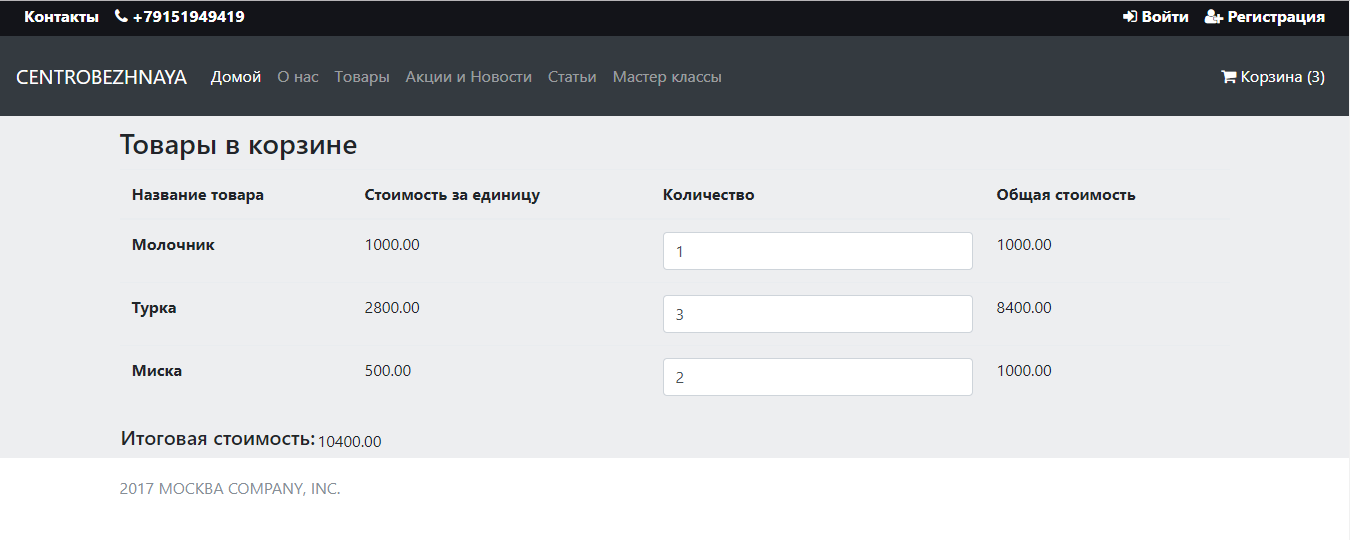


Рисунок 11. Страница заказа

Интерфейс администратора Django

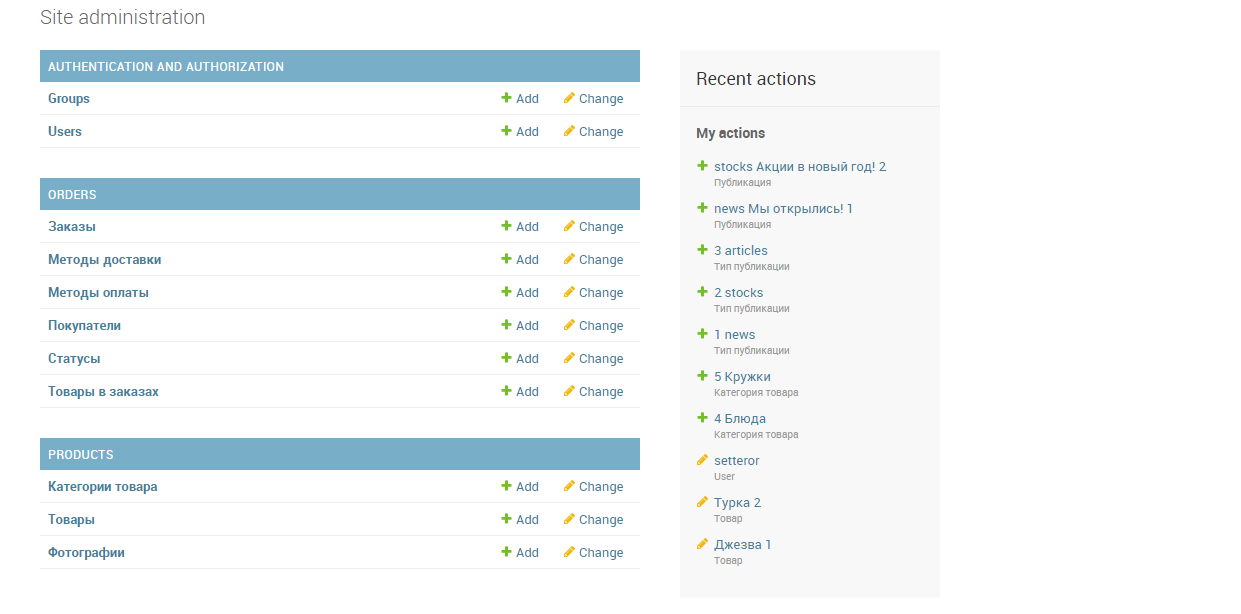


Рисунок 12. Административный раздел

Административный раздел: Заказы

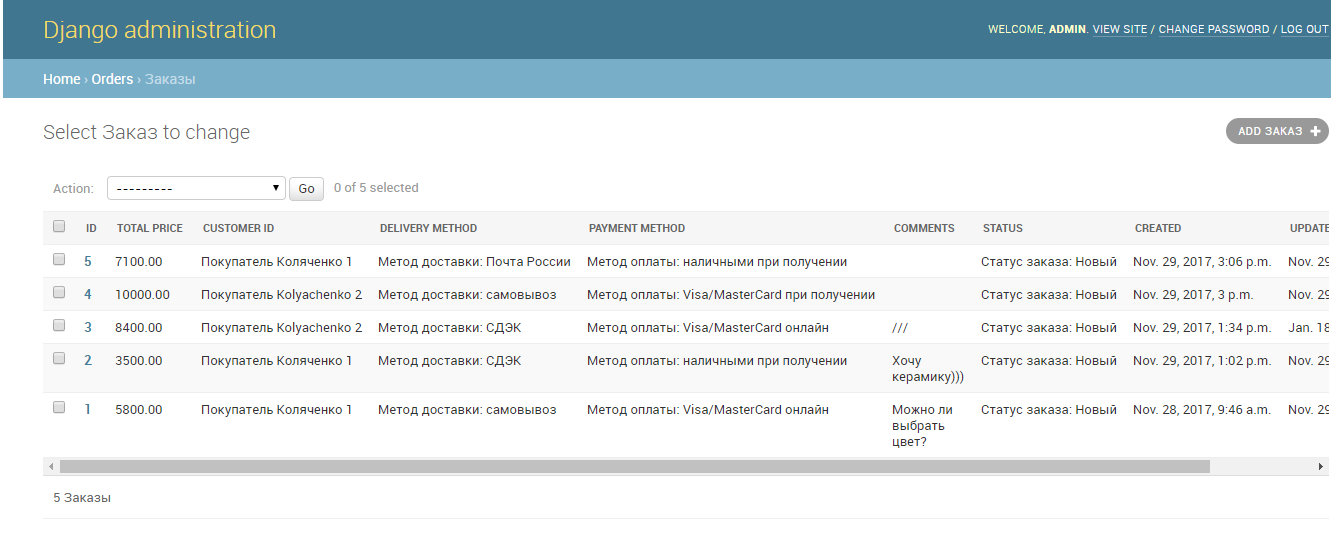


Рисунок 13. Административный раздел

Административный раздел: Редактирование заказа

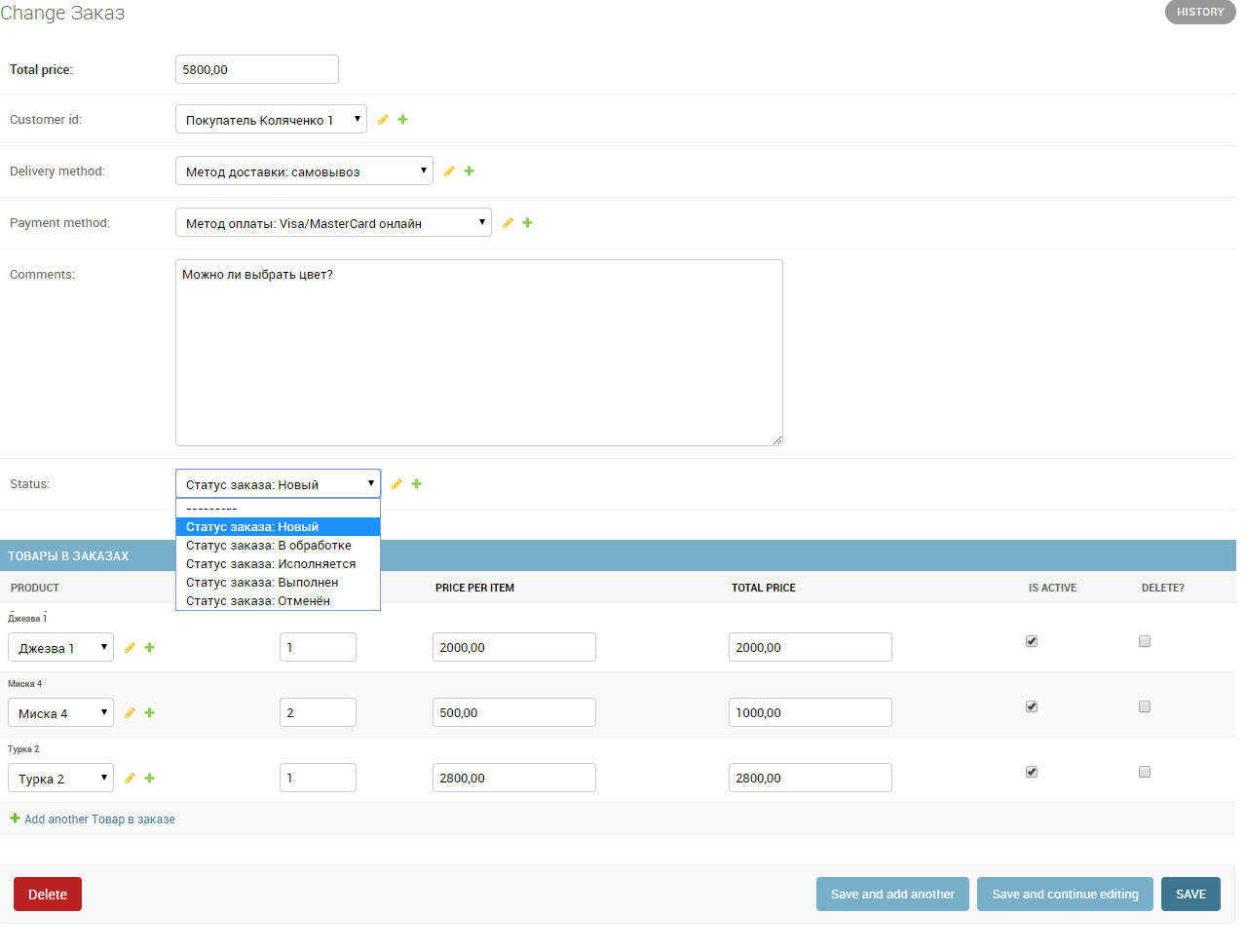


Рисунок 14. Административный раздел

Административный раздел: Метод доставки

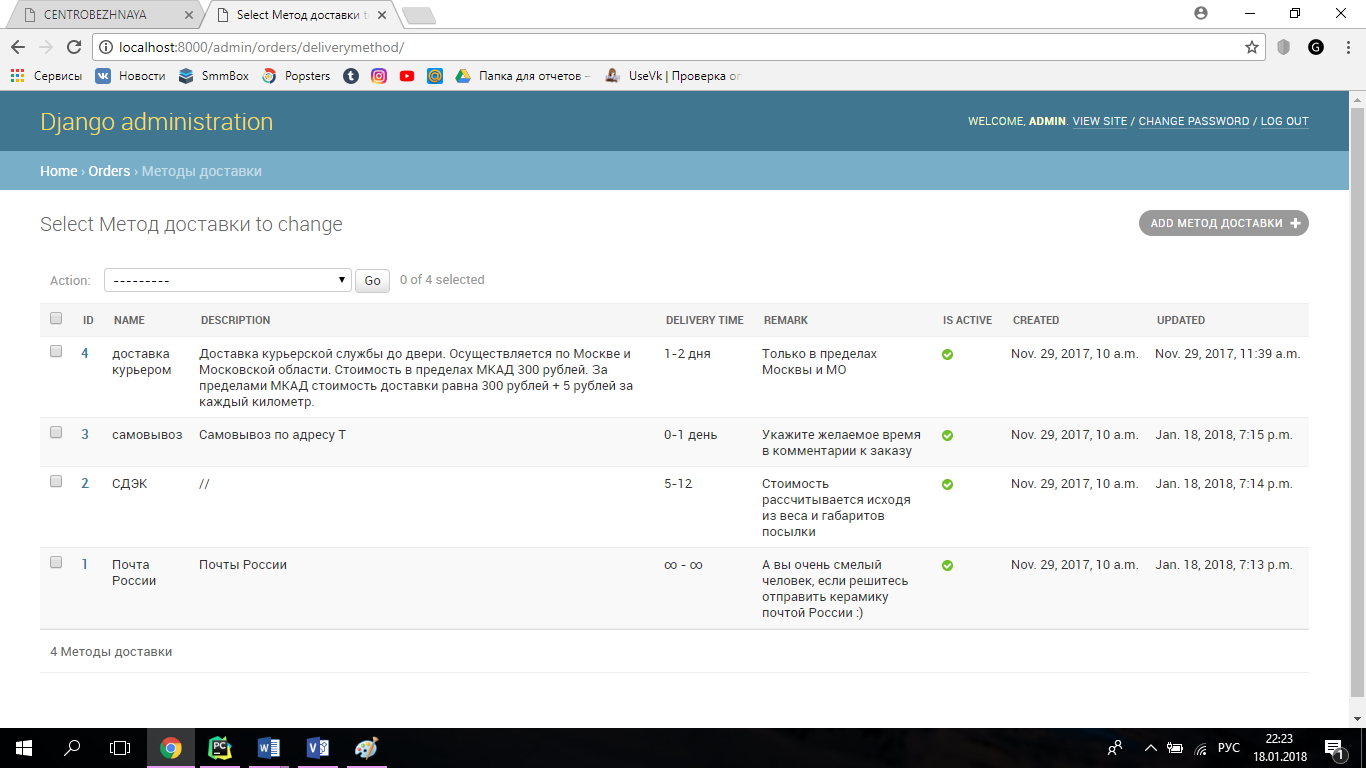


Рисунок 15. Административный раздел

Административный раздел: Метод оплаты

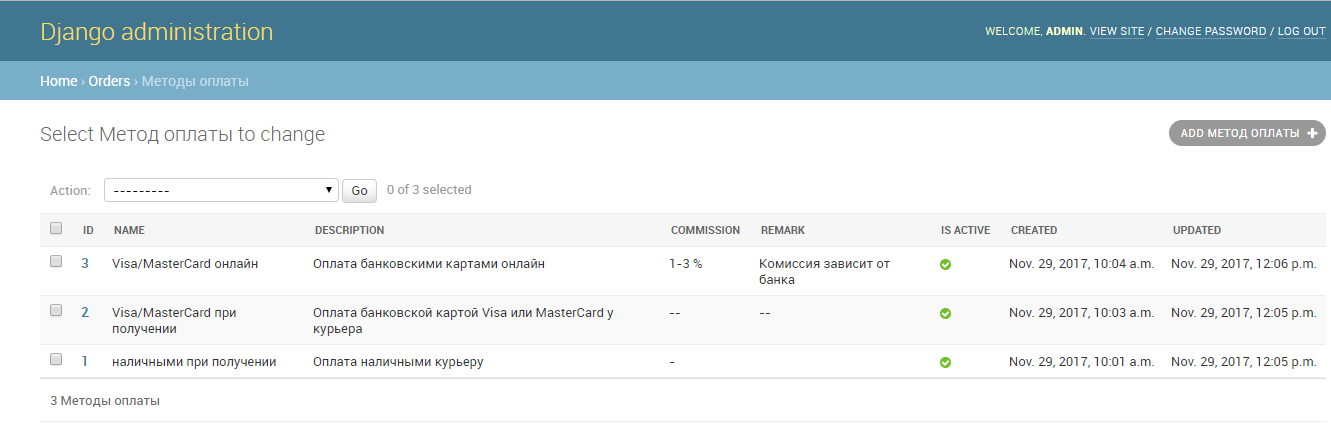


Рисунок 16. Административный раздел

Административный раздел: Статус заказа

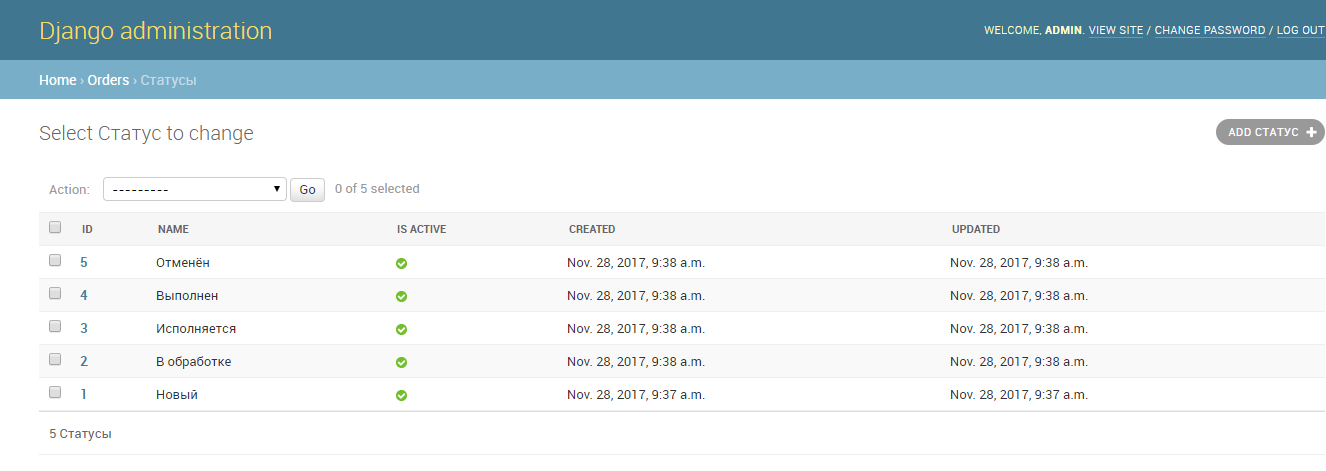


Рисунок 17. Административный раздел

Административный раздел: Товар в заказе

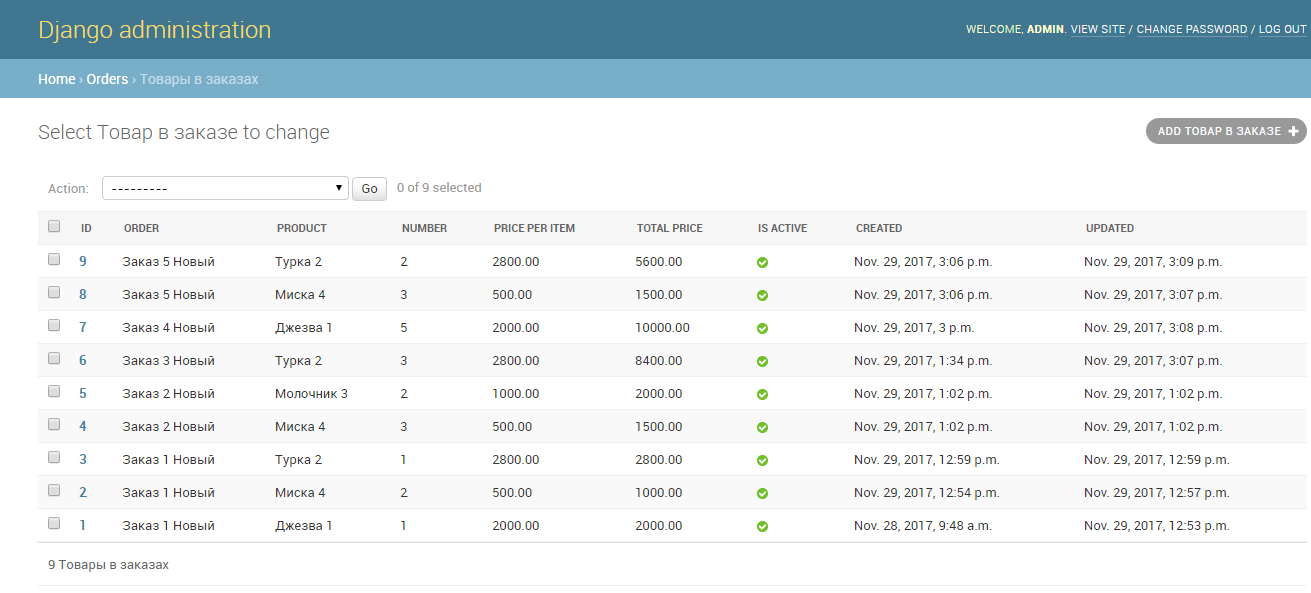


Рисунок 18. Административный раздел

Административный раздел: Категории товаров

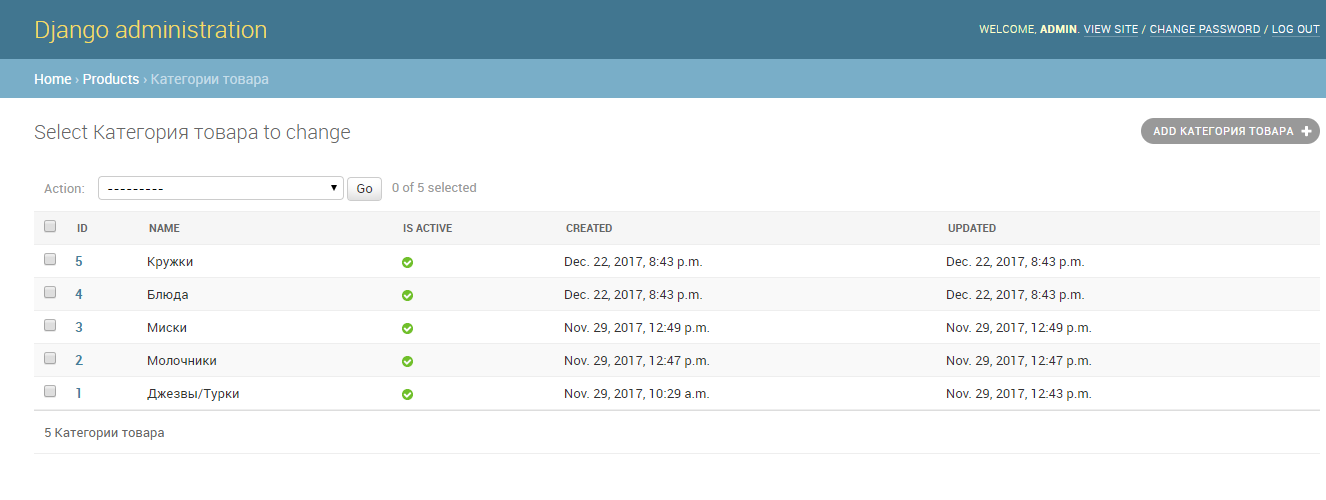


Рисунок 19. Административный раздел

Административный раздел: Добавлении категории товара

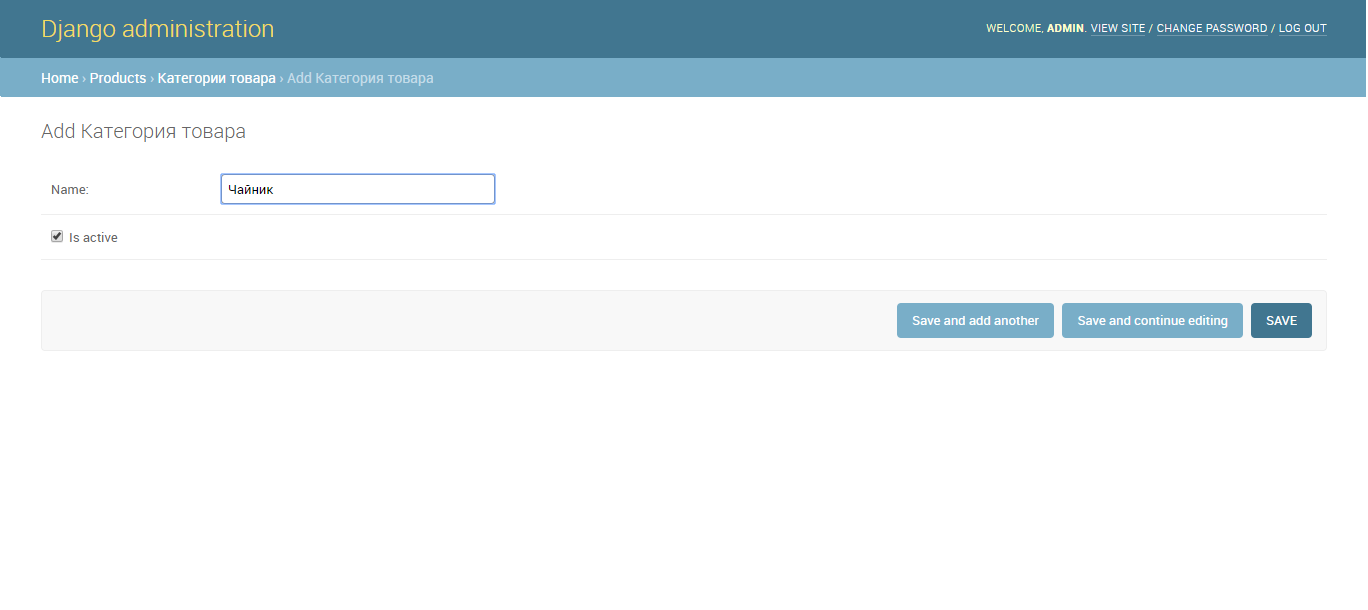


Рисунок 20. Административный раздел

Административный раздел: Товары

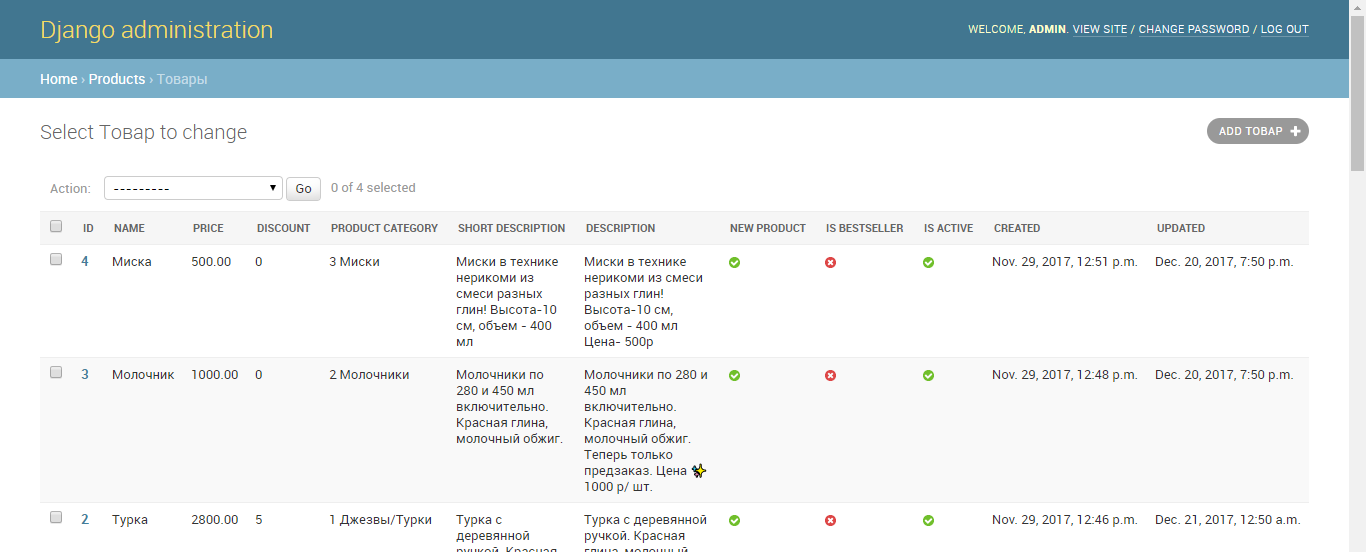


Рисунок 21. Административный раздел

Административный товар: Добавление товара

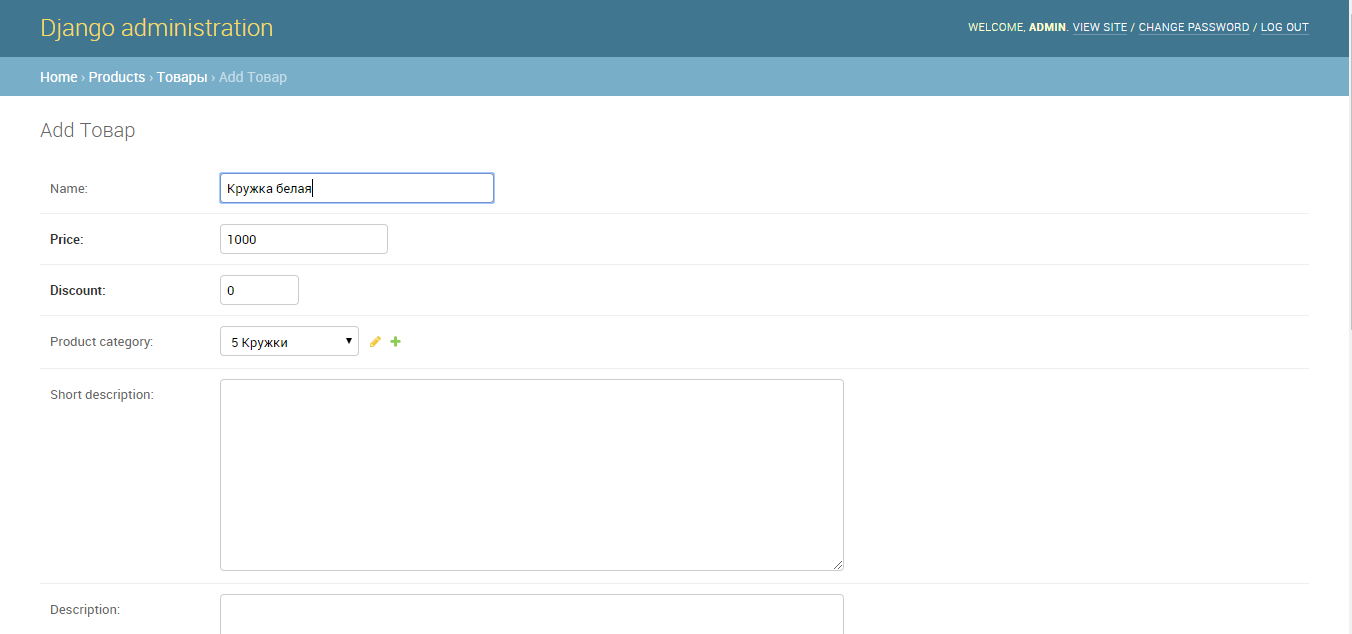


Рисунок 22. Административный раздел

Административный раздел: Фотографии товаров

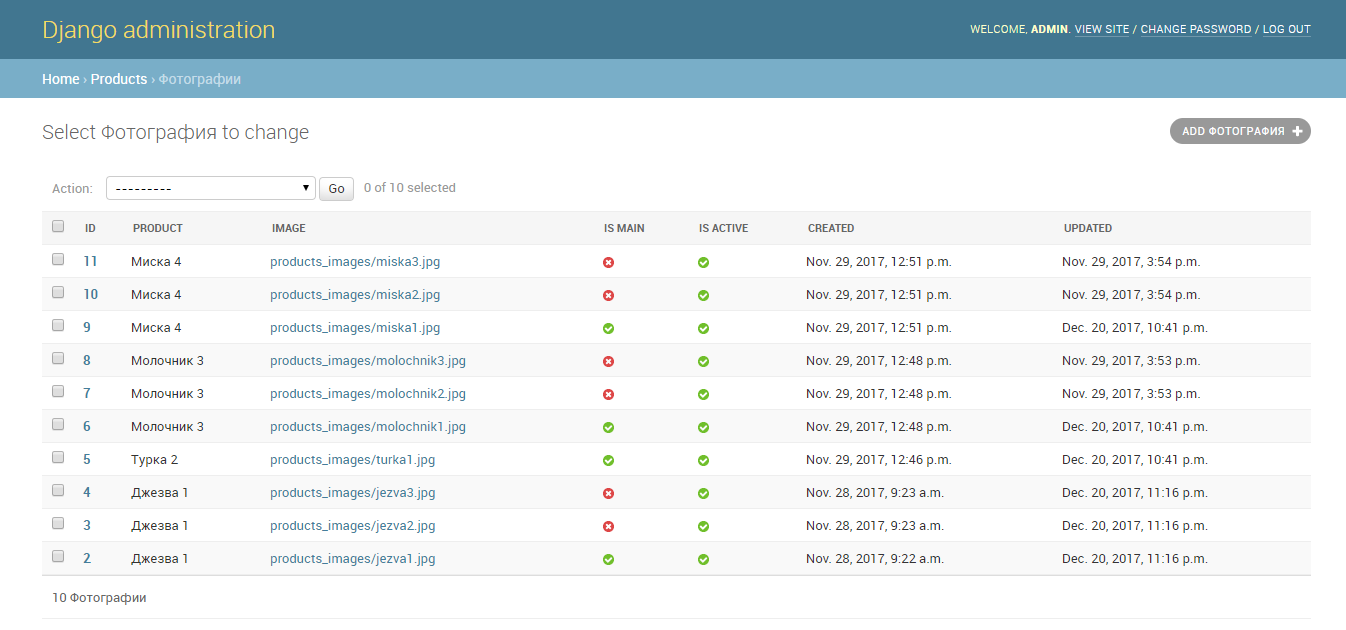


Рисунок 23.Административный раздел

Заключение

В результате проделанной работы было создано web-приложение, согласно требованиям, описанным в техническом задании.

Разработан пользовательский интерфейс при помощи инструментов HTML5, Bootstrap 4, технологии AJAX, библиотеки jQuery и языков СSS, JS.

Улучшены навыки работы в интегрированной среде разработки для языка программирования Python, получены навыки работы с Django - фреймворком для веб-приложений на языке Python.

Для создания приложения была использована интегрированная среда разработки PyCharm.

Таким образом, данное Web-приложения позволяет автоматизировать процесс покупки товара покупателем, минимизировать телефонные и почтовые контакты с продавцом, увеличить число покупателей. Такие задачи и были поставлены Заказчиком.

Приложение

Приложение 1. Исходный код

Settings.py

*"""***import** os  
  
*# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)*BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
  
  
*# Quick-start development settings - unsuitable for production  
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/  
  
# SECURITY WARNING: keep the secret key used in production secret!*SECRET\_KEY = **'+tvn-r1t$=0%\_fv&s^gr&==g^b+4c+k^)n))b%5jgt)0++\*ofy'***# SECURITY WARNING: don't run with debug turned on in production!*DEBUG = True  
  
ALLOWED\_HOSTS = []  
  
  
*# Application definition*INSTALLED\_APPS = [  
 **'django.contrib.admin'**,  
 **'django.contrib.auth'**,  
 **'django.contrib.contenttypes'**,  
 **'django.contrib.sessions'**,  
 **'django.contrib.messages'**,  
 **'django.contrib.staticfiles'**,  
 **'landing'**,  
 **'products'**,  
 **'orders'**,  
 **'login\_auth'**,  
 **'publications'**,  
]  
  
MIDDLEWARE = [  
 **'django.middleware.security.SecurityMiddleware'**,  
 **'django.contrib.sessions.middleware.SessionMiddleware'**,  
 **'django.middleware.common.CommonMiddleware'**,  
 **'django.middleware.csrf.CsrfViewMiddleware'**,  
 **'django.contrib.auth.middleware.AuthenticationMiddleware'**,  
 **'django.contrib.messages.middleware.MessageMiddleware'**,  
 **'django.middleware.clickjacking.XFrameOptionsMiddleware'**,  
]  
  
ROOT\_URLCONF = **'test\_project.urls'**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'**,  
  
 **'orders.context\_processors.getting\_basket\_info'**,  
 ],  
 },  
 },  
]  
  
WSGI\_APPLICATION = **'test\_project.wsgi.application'***# Database  
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases*DATABASES = {  
 **'default'**: {  
 **'ENGINE'**: **'django.db.backends.sqlite3'**,  
 **'NAME'**: os.path.join(BASE\_DIR, **'db.sqlite3'**),  
 }  
}  
  
  
*# Password validation  
# https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators*AUTH\_PASSWORD\_VALIDATORS = [  
 {  
 **'NAME'**: **'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator'**,  
 },  
 {  
 **'NAME'**: **'django.contrib.auth.password\_validation.MinimumLengthValidator'**,  
 },  
 {  
 **'NAME'**: **'django.contrib.auth.password\_validation.CommonPasswordValidator'**,  
 },  
 {  
 **'NAME'**: **'django.contrib.auth.password\_validation.NumericPasswordValidator'**,  
 },  
]  
  
  
*# Internationalization  
# https://docs.djangoproject.com/en/1.11/topics/i18n/*LANGUAGE\_CODE = **'en-us'**TIME\_ZONE = **'UTC'**USE\_I18N = True  
  
USE\_L10N = True  
  
USE\_TZ = True  
  
  
*# Static files (CSS, JavaScript, Images)  
# https://docs.djangoproject.com/en/1.11/howto/static-files/*STATIC\_URL = **'/static/'**STATICFILES\_DIRS = (  
 os.path.join(BASE\_DIR, **"static"**,**"static\_dev"**),  
)  
  
STATIC\_ROOT = os.path.join(BASE\_DIR, **"static"**,**"static\_prod"**)  
  
MEDIA\_URL = **'/media/'**MEDIA\_ROOT = os.path.join(BASE\_DIR, **"static"**, **"media"**)

Urls.py

**from** django.conf.urls **import** url, include  
**from** django.contrib **import** admin  
**from** django.conf **import** settings  
**from** django.conf.urls.static **import** static  
  
urlpatterns = [  
 url(**r'^admin/'**, admin.site.urls),  
 url(**r'^'**, include(**'login\_auth.urls'**)),  
 url(**r'^'**, include(**'landing.urls'**)),  
 url(**r'^'**, include(**'products.urls'**)),  
 url(**r'^'**, include(**'orders.urls'**)),  
 url(**r'^'**, include(**'publications.urls'**)),  
 ] \  
 + static(settings.STATIC\_URL, document\_root=settings.STATIC\_ROOT) \  
 + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

wsgi.py

**import** os  
  
**from** django.core.wsgi **import** get\_wsgi\_application  
  
os.environ.setdefault(**"DJANGO\_SETTINGS\_MODULE"**, **"test\_project.settings"**)  
  
application = get\_wsgi\_application()

orders/admin.py

**from** django.contrib **import** admin  
**from** .models **import** \*  
  
  
**class** ProductInOrderInline(admin.TabularInline):  
 model = ProductInOrder  
 extra = 0  
  
  
**class** CustomerAdmin(admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** Customer.\_meta.fields]  
  
 **class** Meta:  
 model = Customer  
  
  
admin.site.register(Customer, CustomerAdmin)  
  
  
**class** DeliveryMethodAdmin(admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** DeliveryMethod.\_meta.fields]  
  
 **class** Meta:  
 model = DeliveryMethod  
  
  
admin.site.register(DeliveryMethod, DeliveryMethodAdmin)  
  
  
**class** PaymentMethodAdmin(admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** PaymentMethod.\_meta.fields]  
  
 **class** Meta:  
 model = PaymentMethod  
  
  
admin.site.register(PaymentMethod, PaymentMethodAdmin)  
  
  
**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)  
  
  
**class** ProductInBasketAdmin(admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** ProductInBasket.\_meta.fields]  
  
 **class** Meta:  
 model = ProductInBasket  
  
  
admin.site.register(ProductInBasket, ProductInBasketAdmin)

orders/apps.py

**from** django.apps **import** AppConfig  
  
  
**class** LandingConfig(AppConfig):  
 name = **'landing'**

orders/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)  
 products\_total\_numb = products\_in\_basket.count()  
  
 **return** locals()

orders/forms.py

**from** django **import** forms  
**from** .models **import** \*

orders/models.py

**from** django.db **import** models  
**from** products.models **import** Product  
**from** django.db.models.signals **import** post\_save  
  
  
**class** Customer(models.Model):  
 customer\_login = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 customer\_password = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 customer\_firstname = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 customer\_secondname = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 customer\_email = models.EmailField(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)  
 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 "Покупатель %s %s"** % (self.customer\_secondname, self.id)  
  
 **class** Meta:  
 verbose\_name = **'Покупатель'** verbose\_name\_plural = **'Покупатели'  
  
  
class** DeliveryMethod(models.Model):  
 name = models.CharField(max\_length=24, blank=True, null=True, default=None)  
 description = models.TextField(blank=True, null=True, default=None)  
 delivery\_time = models.CharField(max\_length=24, blank=True, null=True, default=None)  
 remark = 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 "Метод доставки: %s"** % self.name  
  
 **class** Meta:  
 verbose\_name = **'Метод доставки'** verbose\_name\_plural = **'Методы доставки'  
  
  
class** PaymentMethod(models.Model):  
 name = models.CharField(max\_length=48, blank=True, null=True, default=None)  
 description = models.TextField(blank=True, null=True, default=None)  
 commission = models.CharField(max\_length=24, blank=True, null=True, default=None)  
 remark = 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 "Метод оплаты: %s"** % self.name  
  
 **class** Meta:  
 verbose\_name = **'Метод оплаты'** verbose\_name\_plural = **'Методы оплаты'  
  
  
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 заказа: %s"** % self.name  
  
 **class** Meta:  
 verbose\_name = **'Статус'** verbose\_name\_plural = **'Статусы'  
  
  
class** Order(models.Model):  
 total\_price = models.DecimalField(max\_digits=10, decimal\_places=2, default=0) *# total price for all products* customer\_id = models.ForeignKey(Customer, blank=True, null=True, default=None)  
 delivery\_method = models.ForeignKey(DeliveryMethod, blank=True, null=True, default=None)  
 payment\_method = models.ForeignKey(PaymentMethod, blank=True, null=True, default=None)  
 comments = models.TextField(blank=True, null=True, default=None)  
 status = models.ForeignKey(Status, blank=True, null=True, default=None)  
 created = models.DateTimeField(auto\_now\_add=True, auto\_now=False)  
 updated = models.DateTimeField(auto\_now\_add=False, auto\_now=True)  
  
 **def** \_\_str\_\_(self):  
 **return "Заказ %s %s"** % (self.id, self.status.name)  
  
 **class** Meta:  
 verbose\_name = **'Заказ'** verbose\_name\_plural = **'Заказы'  
  
 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)  
 product = models.ForeignKey(Product, blank=True, null=True, default=None)  
 number = models.IntegerField(default=1)  
 price\_per\_item = models.DecimalField(max\_digits=10, decimal\_places=2, default=0)  
 total\_price = models.DecimalField(max\_digits=10, decimal\_places=2, default=0) *#price\*nmb\*discount* 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.product.name, self.product.id)  
  
 **class** Meta:  
 verbose\_name = **'Товар в заказе'** verbose\_name\_plural = **'Товары в заказах'  
  
 def** save(self, \*args, \*\*kwargs):  
 price\_per\_item = self.product.price  
 self.price\_per\_item = price\_per\_item  
 self.total\_price = self.number \* price\_per\_item  
  
 super(ProductInOrder, self).save(\*args, \*\*kwargs)  
  
  
**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)  
 number = models.IntegerField(default=1)  
 price\_per\_item = models.DecimalField(max\_digits=10, decimal\_places=2, default=0)  
 total\_price = models.DecimalField(max\_digits=10, decimal\_places=2, default=0) *#price\*nmb\*discount* 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.product.name, self.product.id)  
  
 **class** Meta:  
 verbose\_name = **'Товар в корзине'** verbose\_name\_plural = **'Товары в корзине'  
  
 def** save(self, \*args, \*\*kwargs):  
 price\_per\_item = self.product.price  
 self.price\_per\_item = price\_per\_item  
 self.total\_price = int(self.number) \* price\_per\_item  
  
 super(ProductInBasket, 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)

orders/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'**),  
]

orders/views.py

**from** django.http **import** JsonResponse  
**from** .models **import** ProductInBasket  
  
  
**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"**)  
 numb = data.get(**"nmb"**)  
 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,  
 defaults={**"number"**: numb})  
 **if not** created:  
 new\_product.number += int(numb)  
 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\_numb = products\_in\_basket.count()  
 return\_dict[**"product\_total\_numb"**] = products\_total\_numb  
  
 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[**"numb"**] = item.number  
 return\_dict[**"products"**].append(product\_dict)  
  
 **return** JsonResponse(return\_dict)

products/admin.py

**from** django.contrib **import** admin  
**from** .models **import** \*  
  
  
**class** ProductImageInline(admin.TabularInline):  
 model = ProductImage  
 extra = 0  
  
  
**class** ProductCategoryAdmin(admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** ProductCategory.\_meta.fields]  
  
 **class** Meta:  
 model = ProductCategory  
  
  
admin.site.register(ProductCategory, ProductCategoryAdmin)  
  
  
**class** ProductAdmin(admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** Product.\_meta.fields]  
 inlines = [ProductImageInline]  
  
 **class** Meta:  
 model = Product  
  
  
admin.site.register(Product, ProductAdmin)  
  
  
**class** ProductImageAdmin(admin.ModelAdmin):  
 list\_display = [field.name **for** field **in** ProductImage.\_meta.fields]  
  
 **class** Meta:  
 model = ProductImage  
  
  
admin.site.register(ProductImage, ProductImageAdmin)

products/apps.py

**from** django.apps **import** AppConfig  
  
  
**class** LandingConfig(AppConfig):  
 name = **'landing'**

products/forms.py

**from** django **import** forms  
**from** .models **import** \*

products/models.py

**from** django.db **import** models  
  
  
**class** ProductCategory(models.Model):  
 name = models.CharField(max\_length=48, 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.id, self.name)  
  
 **class** Meta:  
 verbose\_name = **'Категория товара'** verbose\_name\_plural = **'Категории товара'  
  
  
class** Product(models.Model):  
 name = models.CharField(max\_length=64, blank=True, null=True, default=None)  
 price = models.DecimalField(max\_digits=10, decimal\_places=2, default=0)  
 discount = models.IntegerField(default=0)  
 product\_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)  
 new\_product = models.BooleanField(default=True)  
 is\_bestseller = 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 %s"** % (self.name, self.id)  
  
 **class** Meta:  
 verbose\_name = **'Товар'** verbose\_name\_plural = **'Товары'  
  
  
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 "Фотография %s"** % self.id  
  
 **class** Meta:  
 verbose\_name = **'Фотография'** verbose\_name\_plural = **'Фотографии'**

products/urls.py

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

products/views.py

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

static/static\_dev/css/style.css

**html**, **body** {  
 **height**: 100%;  
 **max-height**: 100%;  
 **background-color**: **#edeef0**;  
}  
  
.**wrapper** {  
 **position**: **relative**;  
 **min-height**: 100%;  
}  
  
.**wrapper-content**{  
 **overflow**: **auto**;  
 **padding-bottom**: 180**px**;  
}  
  
.**navbar-top**{  
 **height**:40**px**;  
 **background-color**: **#131419**;  
}  
  
.**navbar-main**{  
 **height**: 80**px**;  
}  
  
.**content** {  
 **box-sizing**: **border-box**;  
 **min-height**: 100%;  
 **padding-bottom**: 90**px**;  
}  
  
.**top-container** {  
 **padding-top**: 5%;  
}  
  
**h1**{  
 **margin-top**: 15**px**;  
 **margin-bottom**: 15**px**;  
 **text-align**: **center**;  
}  
  
**h4**{  
 **margin-top**: 5**px**;  
 **margin-bottom**: 10**px**;  
}  
  
.**nav-top-link**{  
 **color**: **white**;  
 **font-weight**: 700;  
}  
  
.**nav-link**{  
 **text-align**: **right**;  
}  
  
.**nav-top-link**:**hover**{  
 **color**: **#cce8ff**;  
 **font-weight**: 700;  
}  
  
.**profile-container**{  
 **background-color**: **white**;  
 **border-style**: **dashed**;  
 **margin**: 15**px auto** 15**px auto**;  
 **padding**: 10**px** 10**px** 10**px** 10**px**;  
}  
  
.**general-container** {  
 **font-family**: **'Roboto'**, **sans-serif**;  
 **background**: **url**(**'../img/bg-img1.jpg'**) **no-repeat center center fixed**;  
 **height**: 100%;  
 **text-align**: **center**;  
 **webkit-background-size**: **cover**;  
 **-moz-background-size**: **cover**;  
 **-o-background-size**: **cover**;  
 **background-size**: **cover**;  
 **font-weight**: 100;  
}  
  
.**title-container** {  
 **margin-top**: 240**px**;  
}  
  
.**title** {  
 **color**: **#fffff9**;  
 **font-size**: 50**px**;  
 **text-align**: **center**;  
 **font-weight**: 300;  
}  
  
.**brand-title** {  
 **color**: **#fffff9**;  
 **font-size**: 50**px**;  
 **text-align**: **center**;  
 **font-weight**: 700;  
}  
  
.**product-item** {  
 **text-align**: **center**;  
 **height**: 500**px**;  
 **background**: **white**;  
 **position**: **relative**;  
 **margin-top**: 20**px**;  
 **margin-bottom**: 20**px**;  
 **border**: 1**px solid lightgray**;  
 **padding**: 5**px** 0 5**px** 0;  
}  
  
.**add-to-cart-btn** {  
 **position**: **absolute**;  
 **bottom**: 12**px**;  
 **left**: 0;  
 **right**: 0;  
}  
  
.**product-thumbnails-img**{  
 **margin**: 0 5**px** 0 5**px**;  
}  
  
.**slider**{  
 **margin-bottom**: 20**px**;  
}  
  
.**product-short-desc**{  
 **margin**: 5**px**;  
}  
  
.**product-short-price**{  
 **position**: **absolute**;  
 **bottom**: 50**px**;  
 **left**: 0;  
 **right**: 0;  
}  
  
.**product-discount**{  
 **position**: **absolute**;  
 **background**: **#5acc5a**;  
 **top**: 40%;  
 **left**: 0;  
 **width**: 40%;  
 **color**: **white**;  
 **font-weight**: 700;  
}  
  
.**footer** {  
 **position**: **relative**;  
 **margin-top**: -180**px**;  
 **width**: 100%;  
 **height**: 180**px**; */\* Set the fixed height of the footer here \*/* **line-height**: 60**px**; */\* Vertically center the text there \*/* **background-color**: **white**;  
}  
  
.**product-description-tabs**{  
 **margin-top**: 15**px**;  
 **margin-bottom**: 30**px**;  
 **padding-bottom**: 15**px**;  
 **border-bottom**: 1**px solid lightgray**;  
}  
  
.**product-page-img**{  
 **margin**: 15**px** 0 15**px** 0;  
}  
  
.**product-page-price**{  
 **padding**: 15**px** 0 15**px** 0;  
 **margin**: 15**px** 0 15**px** 0;  
 **border-bottom**: 1**px solid lightgray**;  
}  
  
.**nav-product-page**{  
 **padding-top**: 15**px**;  
}  
  
.**btn-page-buy**{  
 **margin-left**: 5**px**;  
}  
  
.**product-page-form-buy**{  
 **padding-top**: 10**px**;  
 **padding-bottom**: 20**px**;  
 **margin**: 15**px** 0 15**px** 0;  
 **border-bottom**: 1**px solid lightgray**;  
}  
  
.**form-signin** {  
 **max-width**: 330**px**;  
 **padding**: 15**px**;  
 **margin**: 0 **auto**;  
}  
.**form-signin** .**form-signin-heading**,  
.**form-signin** .**checkbox** {  
 **margin-bottom**: 10**px**;  
}  
.**form-signin** .**checkbox** {  
 **font-weight**: 400;  
}  
.**form-signin** .**form-control** {  
 **position**: **relative**;  
 **box-sizing**: **border-box**;  
 **height**: **auto**;  
 **padding**: 10**px**;  
 **font-size**: 16**px**;  
}  
.**form-signin** .**form-control**:**focus** {  
 **z-index**: 2;  
}  
.**form-signin input**[**type**=**"email"**] {  
 **margin-bottom**: -1**px**;  
 **border-bottom-right-radius**: 0;  
 **border-bottom-left-radius**: 0;  
}  
.**form-signin input**[**type**=**"password"**] {  
 **margin-bottom**: 10**px**;  
 **border-top-left-radius**: 0;  
 **border-top-right-radius**: 0;  
}  
  
.**modal-header**{  
 **text-align**: **center**;  
 **border**: **none**;  
 **color**: **white**;  
 **margin-left**: 5**px**;  
}  
  
.**modal-content**{  
 **background-color**: **black**;  
 **text-align**: **center**;  
 **color**: **white**;  
}  
  
.**close-modal**{  
 **color**: **white**;  
}  
  
.**close-modal**:**hover**{  
 **color**: **#e9e9e9**;  
}  
  
.**main-modal-contacts**{  
 **margin**: 15**px** 0 15**px** 0;  
 **font-size**: 24**px**;  
}  
  
.**fa-modal**{  
 **margin**: 15**px** 0 15**px** 0;  
 **color**: **white**;  
 **font-size**: 48**px**;  
}  
  
.**fa-modal**:**hover**{  
 **margin**: 15**px** 0 15**px** 0;  
 **color**: **#bef0ff**;  
 **font-size**: 48**px**;  
}  
  
**a**.**text**:**active**, */\* активная/посещенная ссылка \*/***a**.**text** {  
 **text-decoration**: **none**;  
 **color**: **#212529**;  
}  
  
**a**.**text**:**hover** {  
 **text-decoration**: **none**;  
 **color**: **#43474b**;  
}  
  
**ul**.**nav li**.**dropdown**:**hover ul**.**dropdown-menu**{  
**display**: **block**;  
**margin**: 0;  
}  
  
**a**.**menu**:**after**, .**dropdown-toggle**:**after** {  
**content**: **none**;  
}  
  
.**basket-container**{  
 **width**: 400**px**;  
}  
  
.**basket-items**{  
 **width**: 100%;  
}**﻿**.**card-columns**{  
 **text-align**: **center**;  
 }

static/static\_dev/js/scripts.js

$(**document**).ready(**function** () {  
 **var** form = $(**'#form\_buying\_product'**);  
 **console**.log(form);  
  
 **function** *basketupdating*(product\_id, numb, is\_delete) {  
 **var** data = {};  
 data.**product\_id** = product\_id;  
 data.**nmb** = numb;  
 **var** csrf\_token = $(**'#form\_buying\_product [name="csrfmiddlewaretoken"]'**).val();  
 data[**"csrfmiddlewaretoken"**] = csrf\_token;  
  
 **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.product\_total\_numb);  
 **if** (data.product\_total\_numb) {  
 $(**'#product\_total\_numb'**).text(**"("** + data.product\_total\_numb + **")"**);  
 console.log(data.products);  
 $(**'.basket-items ul'**).html(**""**);  
 $.each(data.products, **function** (k, v) {  
 $(**'.basket-items ul'**).append(**'<li><h6>'** + v.name + **', '** + v.numb + **'шт. '** + **'по '** + v.price\_per\_item + **' руб '** +  
 **'<a class="delete-item" href=""data-product\_id "'** + v.id +**' ">x</a>'** +  
 **'</h6></li>'**);  
 })  
 }  
 },  
 error: **function** () {  
 console.log(**"ERROR"**)  
 }  
 });  
 }  
  
 form.on(**'submit'**, **function** (e) {  
 e.preventDefault();  
 console.log(**'123rus'**);  
 **var** numb = $(**'#product\_page\_number'**).val();  
 console.log(numb);  
 **var** submite\_btn = $(**'#product\_submit\_btn'**);  
 **var** product\_id = submite\_btn.data(**"product\_id"**);  
 **var** product\_name = submite\_btn.data(**"product\_name"**);  
 **var** product\_price = submite\_btn.data(**"product\_price"**);  
 console.log(product\_id);  
 console.log(product\_name);  
 console.log(product\_price);  
  
 basketupdating(product\_id, numb, is\_delete=**false**)  
  
 });  
 $(document).on(**'click'**, **'.delete-item'**, **function** (e) {  
 e.preventDefault();  
 product\_id = $(**this**).data(**"product\_id"**);  
 numb = none;  
 basketupdating(product\_id, numb, is\_delete=**true**)  
 })  
});

templates/landing/home.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
{% **block content** %}  
 <**div class="content"**>  
 <**div class="slider"**>  
 <**div class="section-top"**>  
 <**div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel"**>  
 <**ol class="carousel-indicators"**>  
 <**li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"**></**li**>  
 <**li data-target="#carouselExampleIndicators" data-slide-to="1"**></**li**>  
 <**li data-target="#carouselExampleIndicators" data-slide-to="2"**></**li**>  
 </**ol**>  
 <**div class="carousel-inner"**>  
 <**div class="carousel-item active"**>  
 <**img class="d-block w-100" src="**{% **static "img/slider4.png"** %}**" alt="First slide"**>  
 </**div**>  
 <**div class="carousel-item"**>  
 <**img class="d-block w-100" src="**{% **static "img/slider4.png"** %}**" alt="Second slide"**>  
 </**div**>  
 <**div class="carousel-item"**>  
 <**img class="d-block w-100" src="**{% **static "img/slider4.png"** %}**" alt="Third slide"**>  
 </**div**>  
 </**div**>  
 <**a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev"**>  
 <**span class="carousel-control-prev-icon" aria-hidden="true"**></**span**>  
 <**span class="sr-only"**>Предыдущий</**span**>  
 </**a**>  
 <**a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next"**>  
 <**span class="carousel-control-next-icon" aria-hidden="true"**></**span**>  
 <**span class="sr-only"**>Следующий</**span**>  
 </**a**>  
 </**div**>  
 </**div**>  
 </**div**>  
 <**div class="section"**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col-lg-12"**>  
 <**h1**>  
 Товары  
 </**h1**>  
 </**div**>  
 {% **for product\_image in products\_images** %}  
 <**div class="col"**>  
 <**div class="product-item"**>  
 <**a href="**{% **url 'product' product\_image**.**product**.**id** %}**"**>  
 <**div class="product-thumbnails-img"**>  
 <**img src="**{{ **product\_image**.**image**.**url** }}**" class="img-fluid"**>  
 </**div**>  
 </**a**>  
 {% **if product\_image**.**product**.**discount** %}  
 <**div class="product-discount"**>  
 <**span**>{{ **product\_image**.**product**.**discount** }}%</**span**>  
 </**div**>  
 {% **endif** %}  
 <**a class="text" href="**{% **url 'product' product\_image**.**product**.**id** %}**"**>  
 <**h4**>{{ **product\_image**.**product**.**name** }}</**h4**>  
 </**a**>  
 <**div class="product-short-desc"**>  
 <**p**>  
 {{ **product\_image**.**product**.**short\_description** }}  
 </**p**>  
 </**div**>  
 <**div class="product-short-price"**>  
 <**h4**>{{ **product\_image**.**product**.**price** }} &#8381</**h4**>  
 </**div**>  
 <**div class="add-to-cart-btn"**>  
 <**button class="btn btn-success btn-short-buy"**>  
 Добавить в корзину  
 </**button**>  
 </**div**>  
 </**div**>  
 </**div**>  
 {% **endfor** %}  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

templates/login\_auth/login.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
{% **block content** %}  
 <**h1**>Вход</**h1**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col"**>  
 <**form method="post"**>  
 {% **csrf\_token** %}  
 {{ **form**.**as\_p** }}  
 <**button class="btn btn-primary" type="submit"**>Войти</**button**>  
 </**form**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

templates/login\_auth/logout.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
{% **block content** %}  
 <**h1**>Logged out</**h1**>  
 <**h4 class="text-center align-middle"**>You have logged out!</**h4**>  
{% **endblock** %}

templates/login\_auth/register.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
{% **block content** %}  
 <**h1**>Регистрация</**h1**>  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col"**>  
 <**form method="post"**>  
 {% **csrf\_token** %}  
 {{ **form**.**as\_p** }}  
 <**button class="btn btn-primary" type="submit"**>Зарегистрироваться</**button**>  
 </**form**>  
 </**div**>  
 </**div**>  
 </**div**>  
  
{% **endblock** %}

templates/login\_auth/view\_profile.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
{% **block content** %}  
 <**div class="container profile-container rounded"**>  
 <**h1**>Ваш профиль</**h1**>  
 <**h5 class="profile-info"**>Логин: {{ **user** }}</**h5**>  
 <**h5 class="profile-info"**>Имя: {{ **user**.**first\_name** }}</**h5**>  
 <**h5 class="profile-info"**>Фамилия: {{ **user**.**last\_name** }}</**h5**>  
 <**h5 class="profile-info"**>Email: {{ **user**.**email** }}</**h5**>  
 <**form action="/accounts/profile/edit"**>  
 <**button class="btn btn-secondary"**>Редактировать профиль</**button**>  
 </**form**>  
 </**div**>  
{% **endblock** %}

templates/products/product.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
{% **block content** %}  
 <**div class="container"**>  
 <**div class="row"**>  
 <**div class="col"**>  
 {% **for image\_item in product**.**productimage\_set**.**filter** %}  
 <**div class="product-page-img"**>  
 <**img src="**{{ **image\_item**.**image**.**url** }}**" class="img-fluid"**>  
 </**div**>  
 {% **endfor** %}  
 </**div**>  
 <**div class="col"**>  
 <**h1 class="text-center"**>  
 {{ **product**.**name** }}  
 </**h1**>  
 <**div class="product-page-price"**>  
 <**h4**>  
 Цена: {{ **product**.**price** }} &#8381/шт  
 </**h4**>  
 </**div**>  
 <**form class="form-inline product-page-form-buy" id="form\_buying\_product" action="**{% **url 'basket\_adding'** %}**"**>{% **csrf\_token** %}  
 <**div class="form-group"**>  
 <**div class="form-group"**>  
 <**label for="inputPassword2" class="sr-only"**>Password</**label**>  
 <**input type="number" class="form-control" id="product\_page\_number" placeholder=""**>  
 </**div**>  
 </**div**>  
 <**div class="form-group"**>  
 <**button type="submit" class="btn btn-success btn-page-buy" id="product\_submit\_btn"  
 data-product\_id = "**{{ **product**.**id** }}**"  
 data-product\_name = "**{{ **product**.**name** }}**"  
 data-product\_price = "**{{ **product**.**price** }}**"** >  
 Купить  
 </**button**>  
 </**div**>  
 </**form**>  
 <**ul class="nav nav-tabs nav-product-page" id="myTab" role="tablist"**>  
 <**li class="nav-item"**>  
 <**a class="nav-link active" id="home-tab" data-toggle="tab" href="#description" role="tab"  
 aria-controls="home" aria-selected="true"**>Описание</**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" id="profile-tab" data-toggle="tab" href="#delivery" role="tab"  
 aria-controls="profile" aria-selected="false"**>Доставка</**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" id="contact-tab" data-toggle="tab" href="#characteristics" role="tab"  
 aria-controls="contact" aria-selected="false"**>Характеристики</**a**>  
 </**li**>  
 </**ul**>  
 <**div class="tab-content product-description-tabs" id="myTabContent"**>  
 <**div class="tab-pane fade show active" id="description" role="tabpanel" aria-labelledby="home-tab"**>  
 {{ **product**.**description** }}  
 </**div**>  
 <**div class="tab-pane fade" id="delivery" role="tabpanel" aria-labelledby="profile-tab"**>  
 <**h3**>Как мы делаем доставку</**h3**>  
 </**div**>  
 <**div class="tab-pane fade" id="characteristics" role="tabpanel" aria-labelledby="contact-tab"**>  
 <**p**>Характеристики</**p**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

templates/products/show\_products.html

{% **extends 'base.html'** %}  
{% **load static** %}  
  
{% **block content** %}  
  
{% **endblock** %}

templates/publications/base.html

{% **load static** %}  
  
<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>CENTROBEZHNAYA</**title**>  
  
 *<!-- Latest compiled and minified CSS -->* <**link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css"  
 integrity="sha384-PsH8R72JQ3SOdhVi3uxftmaW6Vc51MKb0q5P2rRUpPvrszuE4W1povHYgTpBfshb" crossorigin="anonymous"**>  
  
 <**link rel="stylesheet" href="**{% **static 'css/font-awesome.min.css'** %}**"**>  
  
  
 *<!-- Custom styles for this template -->* <**link rel="stylesheet" href="**{% **static 'css/style.css'** %}**"**>  
 <**link rel="stylesheet" href="**{% **static 'css/base.css'** %}**"**>  
  
  
</**head**>  
<**body**>  
  
 <**div class="wrapper"**>  
 <**div class="wrapper-content"**>  
 {% **include 'navbar.html'** %}  
  
 {% **block content** %}  
 {% **endblock content** %}  
 </**div**>  
  
  
 </**div**>  
  
  
 {% **include 'footer.html'** %}  
  
  
 <**script src="https://code.jquery.com/jquery-3.2.1.min.js"**></**script**>  
 <**script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js"  
 integrity="sha384-vFJXuSJphROIrBnz7yo7oB41mKfc8JzQZiCq4NCceLEaO4IHwicKwpJf9c9IpFgh"  
 crossorigin="anonymous"**></**script**>  
 <**script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"  
 integrity="sha384-alpBpkh1PFOepccYVYDB4do5UnbKysX5WZXm3XxPqe5iKTfUKjNkCk9SaVuEZflJ"  
 crossorigin="anonymous"**></**script**>  
 <**script src="**{% **static 'js/scripts.js'** %}**"**></**script**>  
  
  
</**body**>  
</**html**>

templates/publications/footer.html

<**footer class="footer"**>  
 <**div class="container"**>  
 <**span class="text-muted"**>2017 MOCKBA COMPANY, INC.</**span**>  
 </**div**>  
</**footer**>

templates/publications/header.html

<**div class="header-top"**>  
 <**div class="row"**>  
 <**div class="col"**>  
 <**a href="#" class="h-contacts"**>Контакты</**a**>  
 </**div**>  
 <**div class="col"**>  
 <**form class="form-inline my-2 my-lg-0"**>  
 <**input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search"**>  
 <**button class="btn btn-outline-success my-2 my-sm-0" type="submit"**>Search</**button**>  
 </**form**>  
 </**div**>  
 <**div class="col"**>  
 <**a href="#" class="h-login"**>Вход</**a**>  
 </**div**>  
 <**div class="col"**>  
 <**a href="#" class="h-fav"**>Избранное</**a**>  
 </**div**>  
 <**div class="col"**>  
 <**a href="#" class="h-basket"**>Корзина</**a**>  
 </**div**>  
 </**div**>  
</**div**>

templates/publications/navbar.html

{% **load static** %}  
  
<**nav class="navbar navbar-expand-lg navbar-top"**>  
 <**ul class="navbar-nav mr-auto"**>  
 <**li class="nav-item nav-top-item"**>  
 <**a class="nav-link nav-top-link modal-link active" href="#" data-toggle="modal"  
 data-target="#contactsModal"**>Контакты</**a**>  
 </**li**>  
 <**li class="nav-item nav-top-item"**>  
 <**a class="nav-link nav-top-link" href="#"**><**i class="fa fa-phone" aria-hidden="true"**></**i**> +79151949419</**a**>  
 </**li**>  
 </**ul**>  
 <**ul class="nav navbar-nav navbar-right"**>  
 {% **if username** %}  
 <**li class="nav-item nav-top-item auth\_user"**>  
 <**a class="nav-link nav-top-link active" href="/logout"**>  
 <**i class="fa fa-sign-out" aria-hidden="true"**></**i**>Выйти ({{ **username** }})</**a**>  
 </**li**>  
 {% **else** %}  
 <**a class="nav-link nav-top-link active" href="/login"**>  
 <**i class="fa fa-sign-in" aria-hidden="true"**></**i**> Войти</**a**>  
 </**li**>  
 {% **endif** %}  
 {% **if username** %}  
 <**li class="nav-item nav-top-item"**>  
 <**a class="nav-link nav-top-link" href="/accounts/profile"**><**i class="fa fa-user-plus"  
 aria-hidden="true"**></**i**>  
 Профиль</**a**>  
 </**li**>  
 {% **else** %}  
 <**li class="nav-item nav-top-item"**>  
 <**a class="nav-link nav-top-link" href="/register"**><**i class="fa fa-user-plus" aria-hidden="true"**></**i**>  
 Регистрация</**a**>  
 </**li**>  
 {% **endif** %}  
 </**ul**>  
</**nav**>  
<**nav class="navbar navbar-expand-lg sticky-top navbar-dark bg-dark navbar-main"**>  
 <**a class="navbar-brand" href="/"**>CENTROBEZHNAYA</**a**>  
 <**button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"  
 aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"**>  
 <**span class="navbar-toggler-icon"**></**span**>  
 </**button**>  
  
 <**div class="collapse navbar-collapse" id="navbarSupportedContent"**>  
 <**ul class="navbar-nav mr-auto"**>  
 <**li class="nav-item active"**>  
 <**a class="nav-link" href="/"**>Домой<**span class="sr-only"**>(current)</**span**></**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="#"**>О нас</**a**>  
 </**li**>  
 <**li class="nav-item dropdown"**>  
 <**a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown"  
 aria-haspopup="true" aria-expanded="false"**>  
 Товары  
 </**a**>  
 <**div class="dropdown-menu" aria-labelledby="navbarDropdown"**>  
 <**a class="dropdown-item" href="/products/cezvi"**>Турки</**a**>  
 <**a class="dropdown-item" href="#"**>Молочники</**a**>  
 <**a class="dropdown-item" href="#"**>Миски</**a**>  
 <**a class="dropdown-item" href="#"**>Блюда</**a**>  
 <**a class="dropdown-item" href="#"**>Кружки</**a**>  
 <**div class="dropdown-divider"**></**div**>  
 <**a class="dropdown-item" href="#"**>Воплотить свою идею!</**a**>  
 </**div**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="/news"**>Акции и Новости</**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="/articles"**>Статьи</**a**>  
 </**li**>  
 <**li class="nav-item"**>  
 <**a class="nav-link" href="#"**>Мастер классы</**a**>  
 </**li**>  
 </**ul**>  
 <**ul class="nav navbar-nav navbar-right"**>  
 <**li class="dropdown basket-container"**>  
 <**a href="#" class="dropdown-toggle nav-link active" data-toggle="dropdown" role="button"  
 aria-haspopup="true"  
 aria-expanded="true"**><**i class="fa fa-shopping-cart" aria-hidden="true"**></**i**> Корзина <**span  
 id="product\_total\_numb"**>  
 {% **if products\_total\_numb** %}  
 ({{ **products\_total\_numb** }})  
 {% **else** %}  
 (0)  
 {% **endif** %}  
 </**span**></**a**>  
 <**div class="dropdown-menu basket-items"**>  
 <**ul class=""**>  
 {% **if products\_total\_numb** %}  
 {% **for product\_in\_basket in products\_in\_basket** %}  
 <**li**><**h6**>{{ **product\_in\_basket**.**product**.**name** }} {{ **product\_in\_basket**.**number** }} шт.  
 по {{ **product\_in\_basket**.**price\_per\_item** }} руб. <**a class="delete-item"  
 href=""data-product\_id="**{{ **product\_in\_basket**.**id** }}**"**>x</**a**>  
 </**h6**></**li**>  
 {% **endfor** %}  
 {% **else** %}  
 <**h4**>Корзина пуста!</**h4**>  
 {% **endif** %}  
 </**ul**>  
 </**div**>  
 </**li**>  
 </**ul**>  
 </**div**>  
</**nav**>  
  
*<!-- Modal -->*<**div class="modal fade" id="contactsModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"  
 aria-hidden="true"**>  
 <**div class="modal-dialog" role="document"**>  
 <**div class="modal-content"**>  
 <**div class="modal-header"**>  
 <**h4 class="modal-title" id="contactsModalLabel"**>Наши контакты</**h4**>  
 <**button type="button" class="close close-modal" data-dismiss="modal" aria-label="Close"**>  
 <**span aria-hidden="true"**>**&times;**</**span**>  
 </**button**>  
 </**div**>  
 <**div class="modal-body"**>  
 <**h5 class="main-modal-contacts"**><**i class="fa fa-phone" aria-hidden="true"**></**i**> +79151949419</**h5**>  
 <**div class="row"**>  
 <**div class="col modal-contacts"**>  
  
 <**a href="https://www.instagram.com/centrobezhnaja/" target="\_blank"**><**i  
 class="fa fa-modal fa-instagram"  
 aria-hidden="true"**></**i**></**a**>  
 </**div**>  
 <**div class="col"**>  
 <**a href="#"**><**i class="fa fa-modal fa-facebook" aria-hidden="true"**></**i**></**a**>  
 </**div**>  
 </**div**>  
 <**div class="row"**>  
 <**div class="col"**>  
 <**a href="#"**><**i class="fa fa-modal fa-vk" aria-hidden="true"**></**i**></**a**>  
 </**div**>  
 <**div class="col"**>  
 <**a href="#"**><**i class="fa fa-modal fa-google-plus" aria-hidden="true"**></**i**></**a**>  
 </**div**>  
 </**div**>  
 <**h5 class="main-modal-contacts"**><**i class="fa fa-envelope-o" aria-hidden="true"**></**i**> example@gmail.com  
 </**h5**>  
 </**div**>  
 </**div**>  
 </**div**>  
</**div**>

Список использованных источников

Форсье, Д., Биссекс, П., & Уэсли, Д. (б.д.). *Django. Разработка веб-приложений на Python.* Символ-Плюс.

Васильев, А. (2017). Python на примерах. Практический курс по программированию. Наука и техника.

Дронов, В. (2016). *Django: практика создания Web-сайтов на Python.* Санкт-Петербург: БХВ-Петербург.

Тахагхогхи, С., & Вильямс, Х. (2007). *Руководство по MySQL.* Русская Редакция.