МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информационных технологий»

**КУРСОВАЯ РАБОТА**

«Создание проекта Интернет-Магазина»

Выполнил: студент гр. БИБ2103 Курчижкина Д.А.

Москва 2022 г.

В создании интернет-магазина использовалась версия Python 3.7, Django 3.9, Pillow 9.3

Создаем новый проект под названием myshop с приложением, называемым shop, открыв терминал и выполнив следующие команды:

django-admin startproject myshop

cd myshop/

django-admin startapp shop

Файл settings.py содержит в себе все настройки проекта, подключение приложений orders, cart, shop см. рис. 1, была также настроена база данных. Также добавили media url для отображения изображений продуктов рис. 2



Рисунок 1 - подключение приложений



Рисунок 2 - Отображение изображений

Файл manage.py показывает модели для интернет-магазина для отображения в базе данных см. рис.3

category : Это ForeignKey модели Category. Это отношение "многие к

одному": продукт относится к одной категории, а категория содержит

несколько продуктов

• name : Название продукта.

• slug : Алиас продукта(его URL).

• image : Изображение продукта.

• description : Необязательное описание для продукта.

• price : Это поле DecimalField. В нем используется десятичное число

Python. Десятичный тип для хранения десятичного числа с

фиксированной точностью. Максимальное число цифр (включая

десятичные разряды) задается с помощью атрибута max\_digits и

десятичных знаков с атрибутом decimal\_places

• stock : Это поле PositiveIntegerField для хранения остатков данного

продукта.

• available : Это булево значение, указывающее, доступен ли продукт или

нет. Позволяет включить/отключить продукт в каталоге.

• created : Это поле хранит дату когда был создан объект.

• updated : В этом поле хранится время последнего обновления объекта.

MEDIA\_URL является базовым URL-адресом, который обслуживает файлы

мультимедиа, загруженные пользователями. MEDIA\_ROOT — это локальный

путь, в котором находятся эти файлы, который мы строим динамически в зависимости от переменной BASE\_DIR.



Рисунок 3 - модель для интернет-магазина

Файл admin.py, чтобы модели были отображены на сайте администрирования см. рис.4, теперь можно создавать продукты на сайте.



Рисунок 4 - файл admin.py

Далее создаем html, css файлы

Base.html

Подключение фавикона представлено на рис. 5

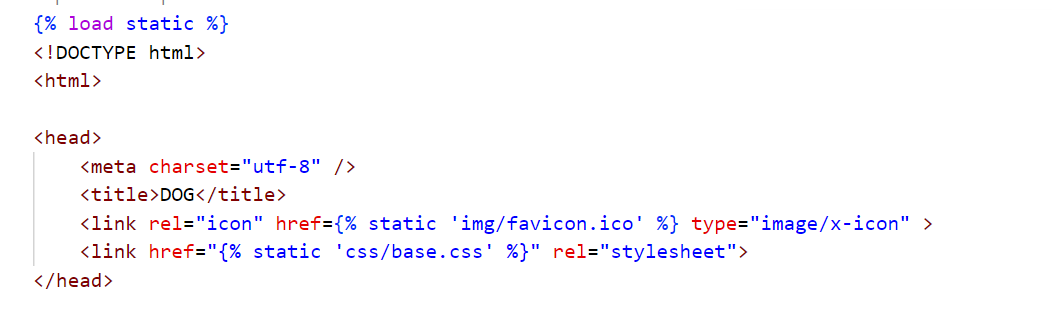


Рисунок 5 – favicon

Подключение фичи падающих снежинок см. рис. 6



Рисунок 6 – Фича

Проверка наличия вещей в корзине см. рис.7

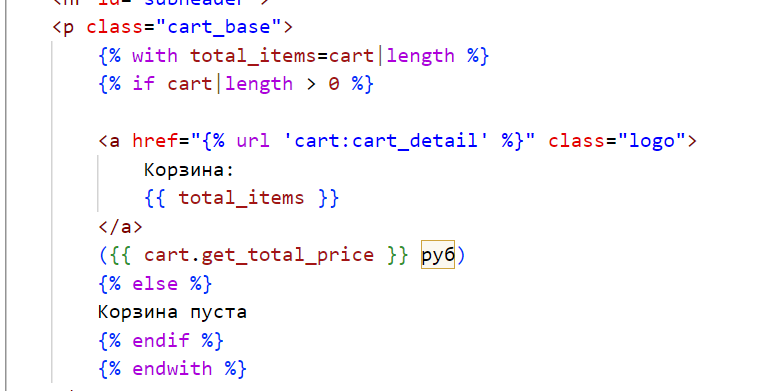


Рисунок 7 - Код для корзины

Если картинка заложена в базе данных, то при помощи цикла есть возможность ее вытащить, если же нет, то будет фотография, говорящая, что картинки нет см. рис.8

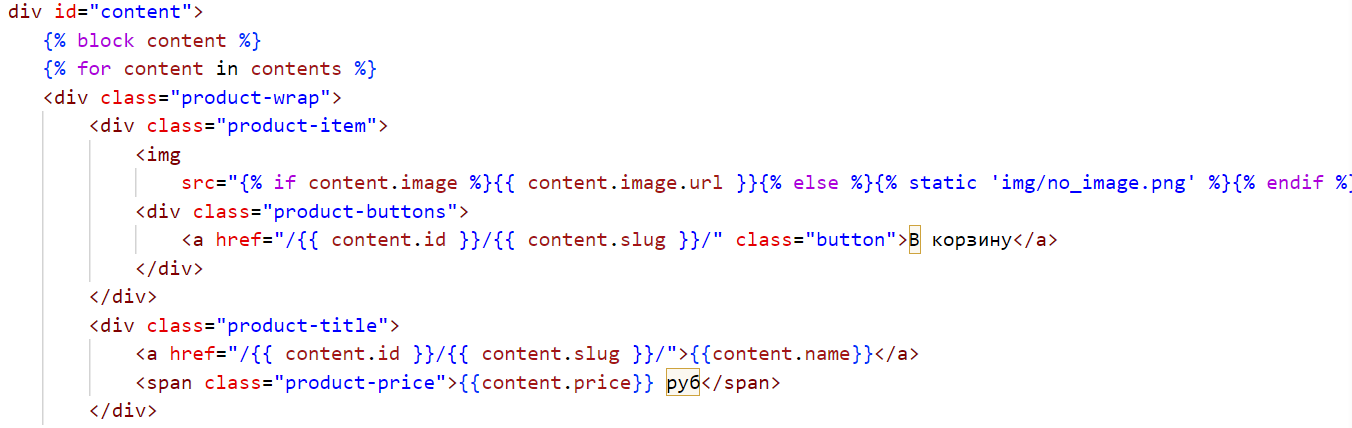


Рисунок 8 - Контент для сайта

Фича на языке JavaScript представлена на рис. 9

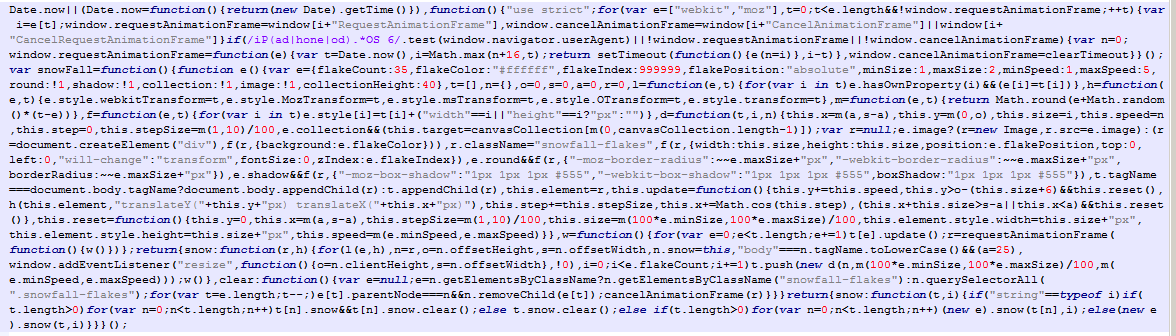


Рисунок 9 – Код фичи

Файл detail.html и list.html дополняет основной файл использует переменную контекста категорий для отображения всех категорий на боковой панели и продуктов для отображения продуктов текущей страницы.

Файл urls.py прописывает адрес конкретной позиции в каталоге, админ-панель, главную страницу см. рис. 10.





Рисунок 10 - Urls.py

Для корзины был подключен файл cart.py, позволяет управлять корзиной, а также дает возможность не добавлять продукт в корзину более одного раза, также для нее был создан шаблон в html.

Модель order

Это модель для хранения сведений о заказе и второй модели для хранения купленных товаров, включая их количество и цену.

Модель Order содержит несколько полей для сведений о клиенте и поле paid, которое по умолчанию имеет значение False. Позже мы будем использовать это поле для различения оплаченных и неоплаченных заказов. Мы также определяем метод get\_total\_cost(), чтобы получить общую стоимость товаров, купленных в этом заказе. Модель OrderItem позволяет хранить продукт, количество и цену, уплаченную за каждый товар. Мы включаем get\_cost() для возврата стоимости товара

Список всех папок представлен на рис. 11

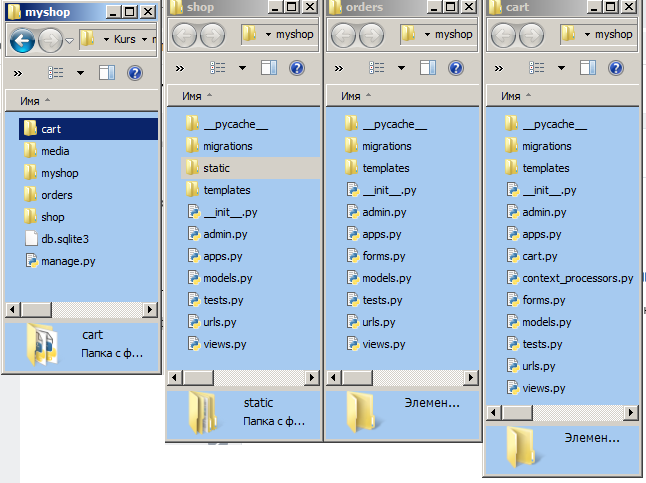


Рисунок 11 - Папки проектов

Сайт представлен далее на рис. 12, 13, 14, 15, 16



Рисунок 12 - Главная страница



Рисунок 13 - Главная страница с категории шарики

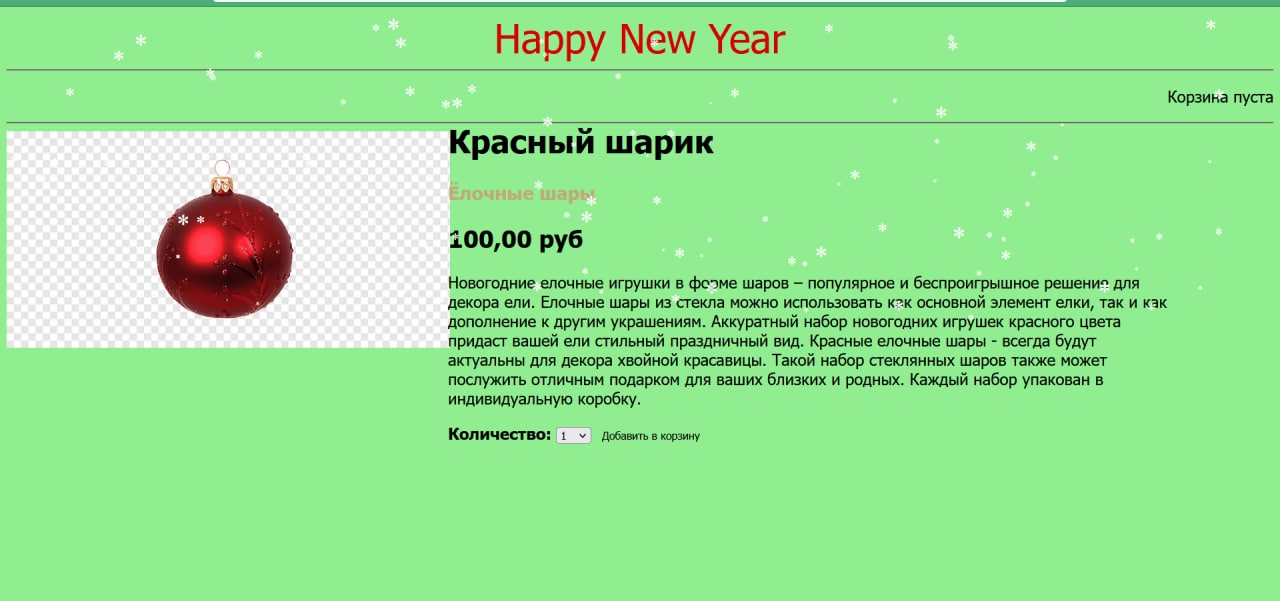


Рисунок 14 - Просмотр данных о товаре

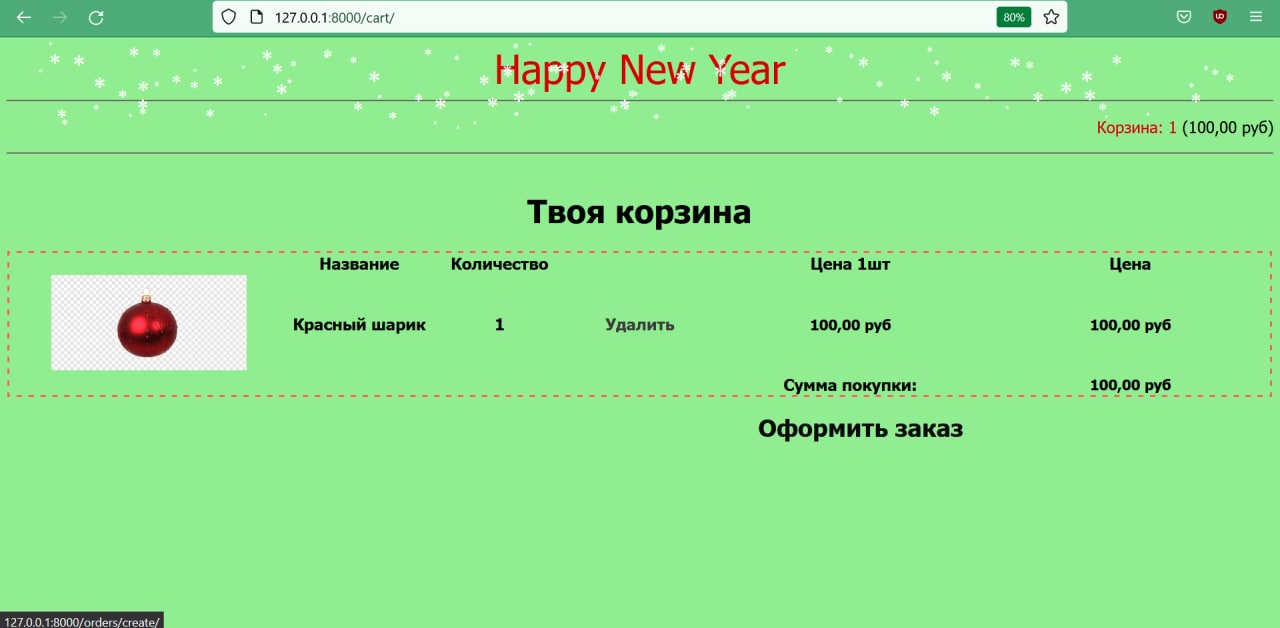


Рисунок 15 - Корзина покупок

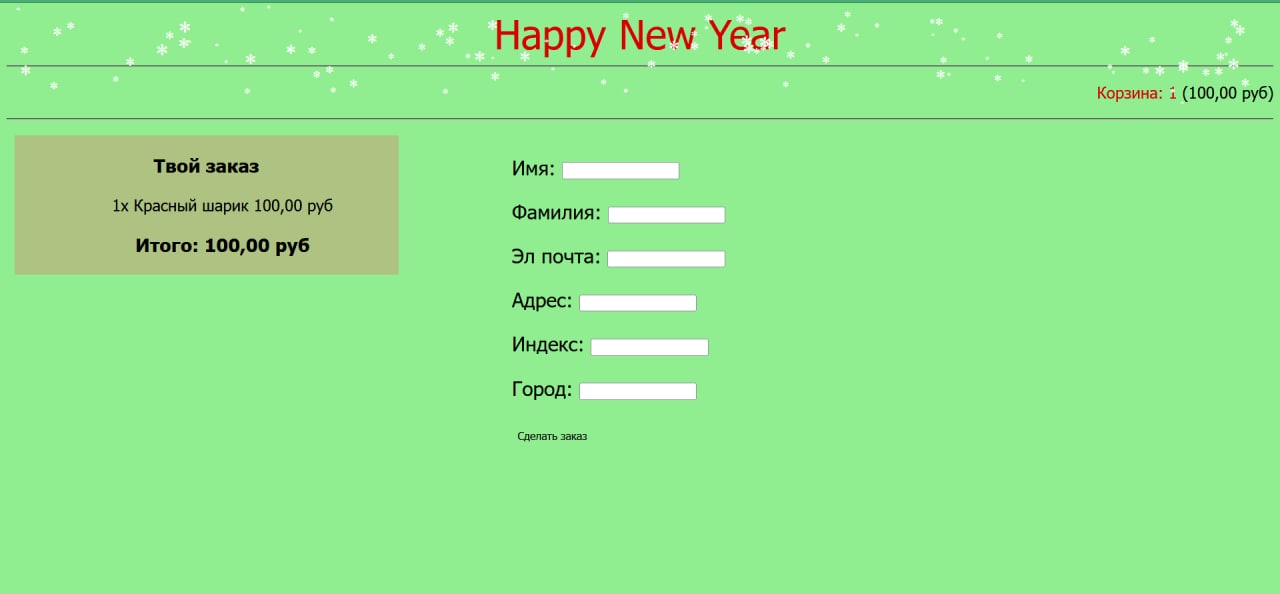


Рисунок 16 - Оформление заказа

Весь код выложен на GitHub по ссылке: https://github.com/Kurchizhkina/Lab\_works-by-TiMP