Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность 09.02.07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

По дисциплине «Технология разработки и защиты баз данных»

Листов: \_\_

|  |  |
| --- | --- |
| Выполнил: студент  Студент группы П50-5-22  Ходжахов Э.Т. | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_А. В. Архангельский  «\_\_\_\_» \_\_\_\_\_\_\_\_ 2025 года |

Москва 2025

ОГЛАВЛЕНИЕ

[ПРАКТИЧЕСКАЯ РАБОТА №1 3](#_Toc197718691)

[ПРАКТИЧЕСКАЯ РАБОТА №2 10](#_Toc197718692)

[ПРАКТИЧЕСКАЯ РАБОТА №3 20](#_Toc197718693)

[ПРАКТИЧЕСКАЯ РАБОТА №4 23](#_Toc197718694)

[ПРАКТИЧЕСКАЯ РАБОТА №5 28](#_Toc197718695)

# ПРАКТИЧЕСКАЯ РАБОТА №1

Цель работы: Реализовать структуру маршрутизации в соответствии с прикрепленным шаблоном. В страницах, где предполагается наличие идентификаторов (просмотр товара, изменение товара) или передача других данных учесть это и реализовать маршрут в соответствии с наличием и применением этих данных.

Ход работы:

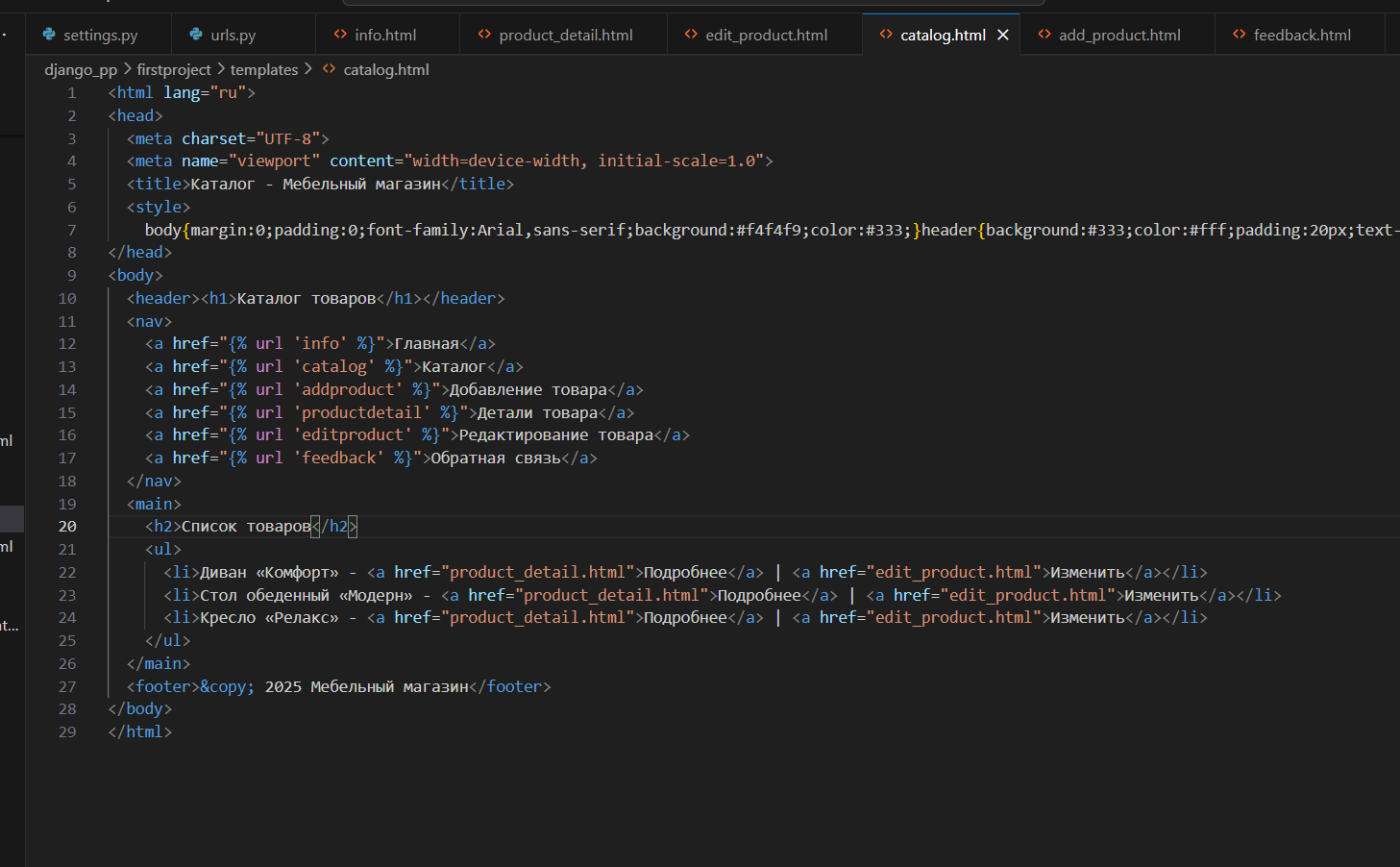


Рисунок – Пример верстки для каталога.

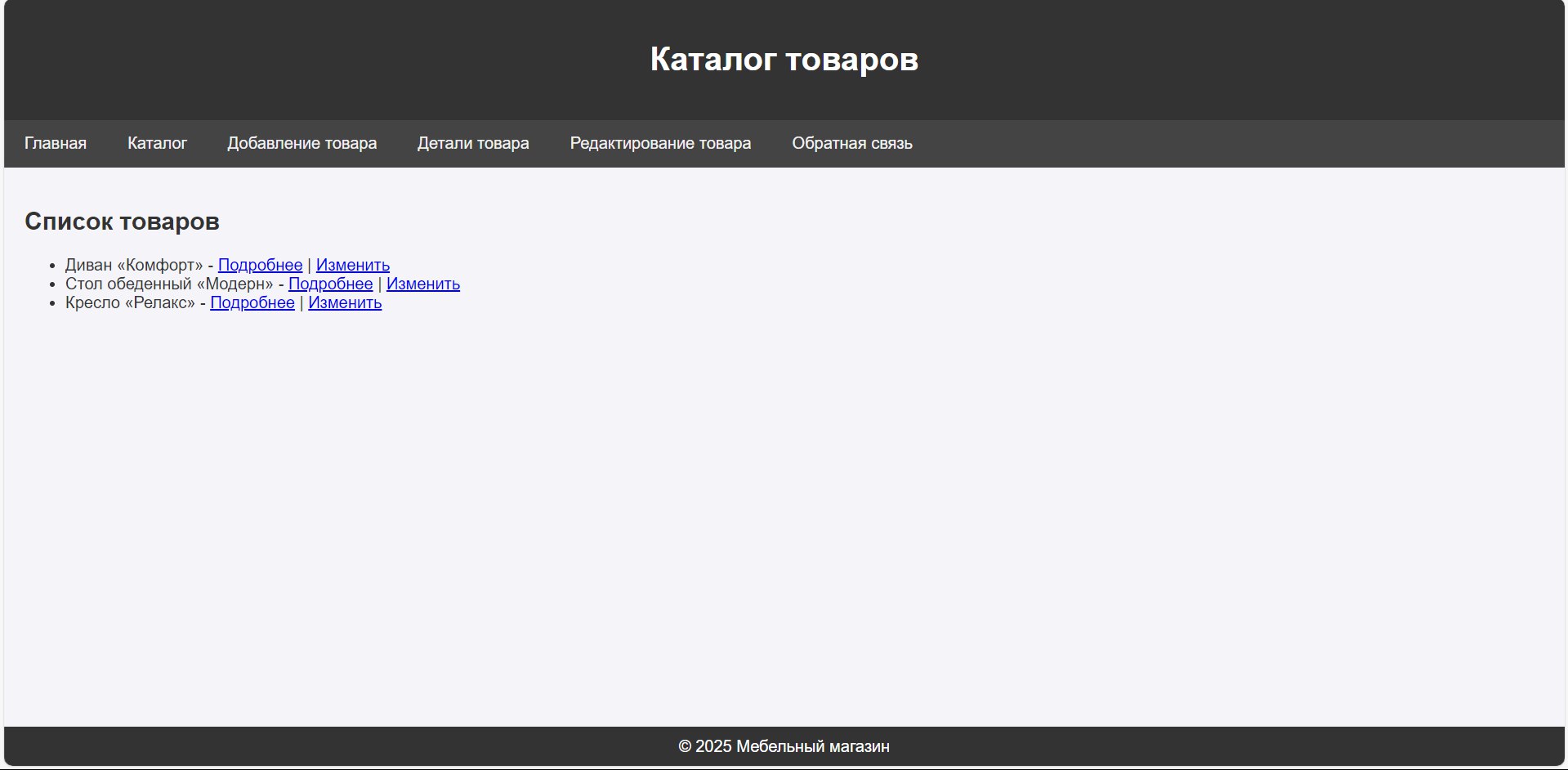


Рисунок – Пример навигации по сайту.

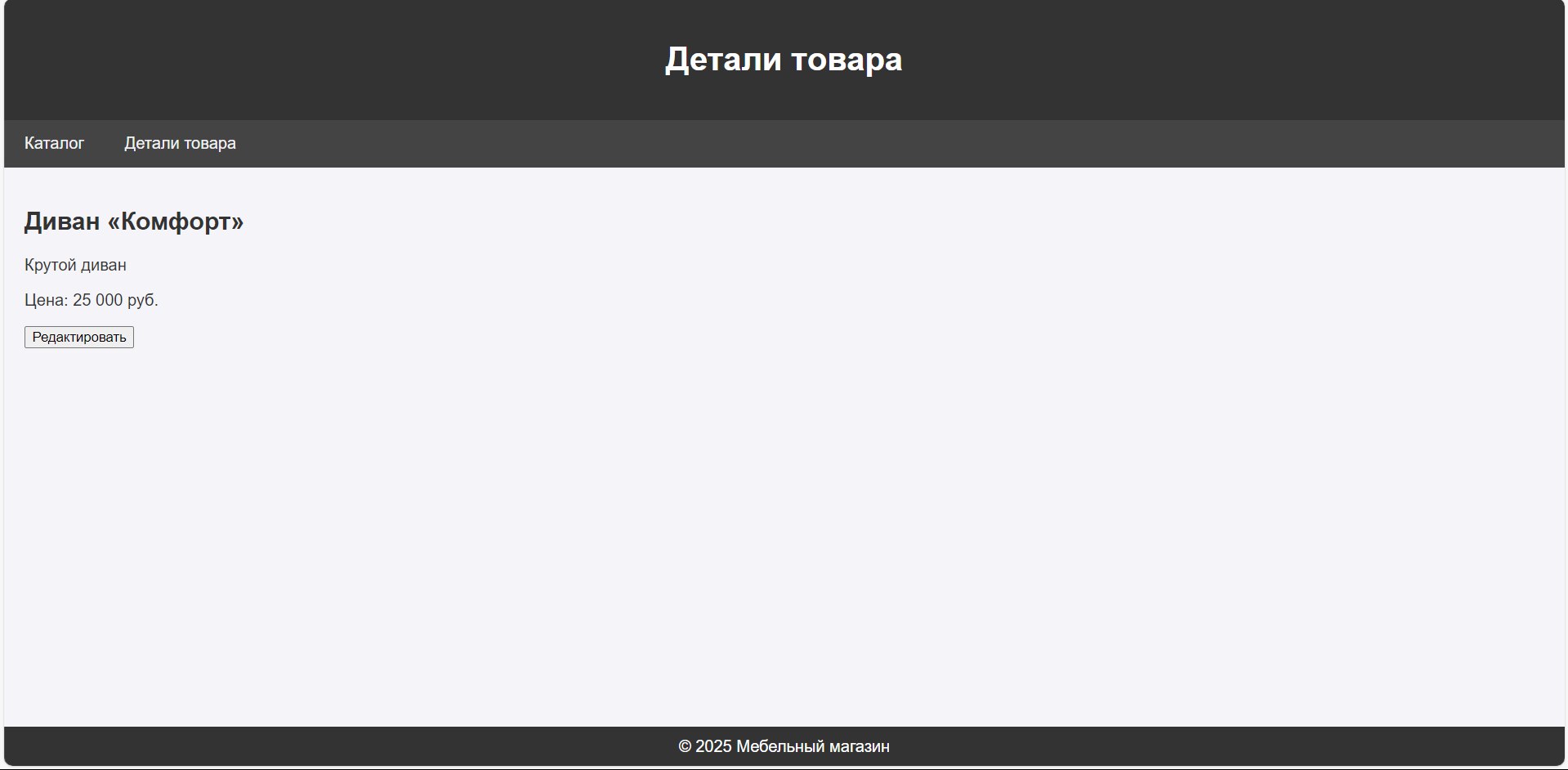


Рисунок – Второй пример навигации по сайту.

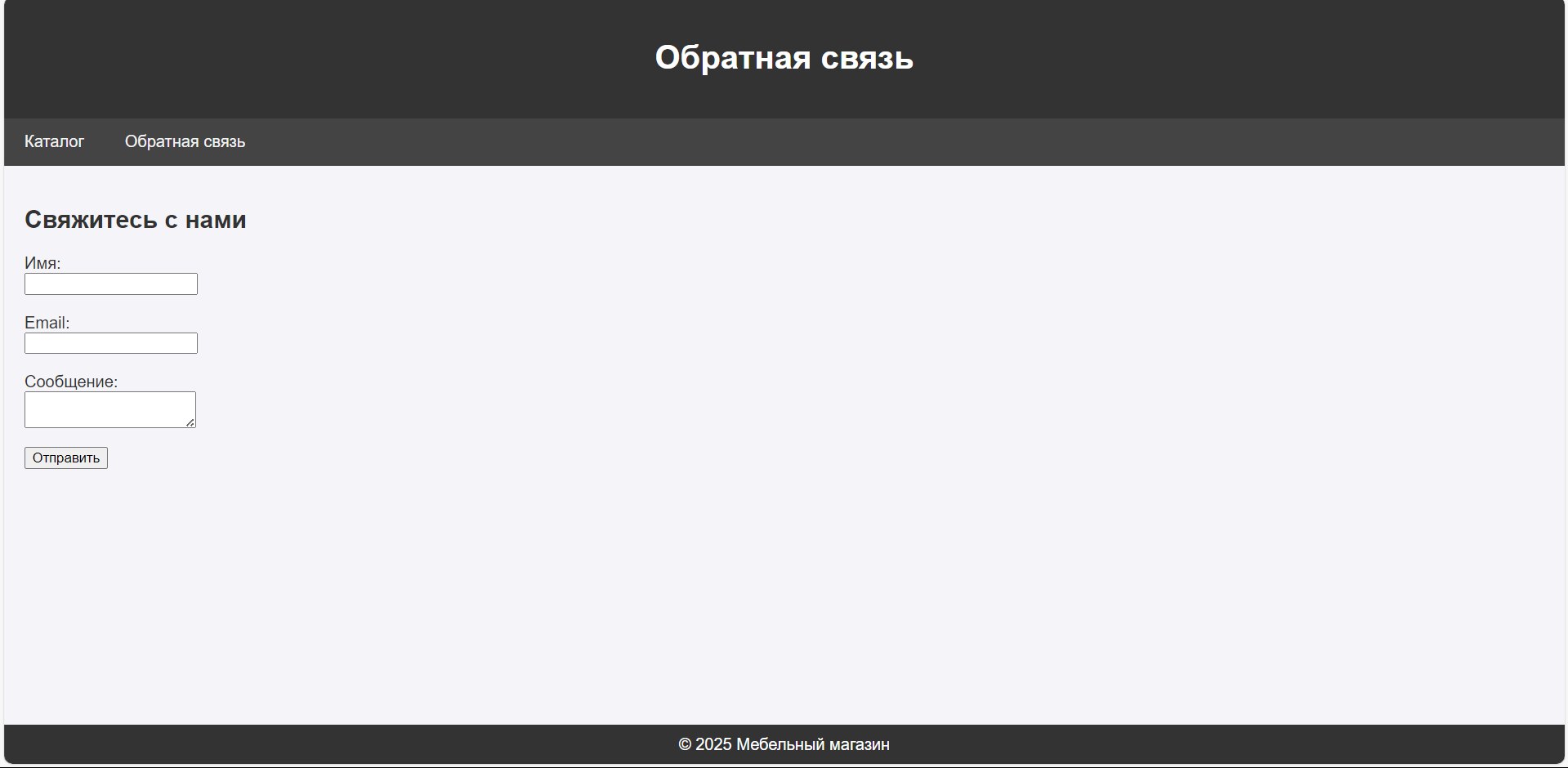


Рисунок – Третий пример навигации по сайту.

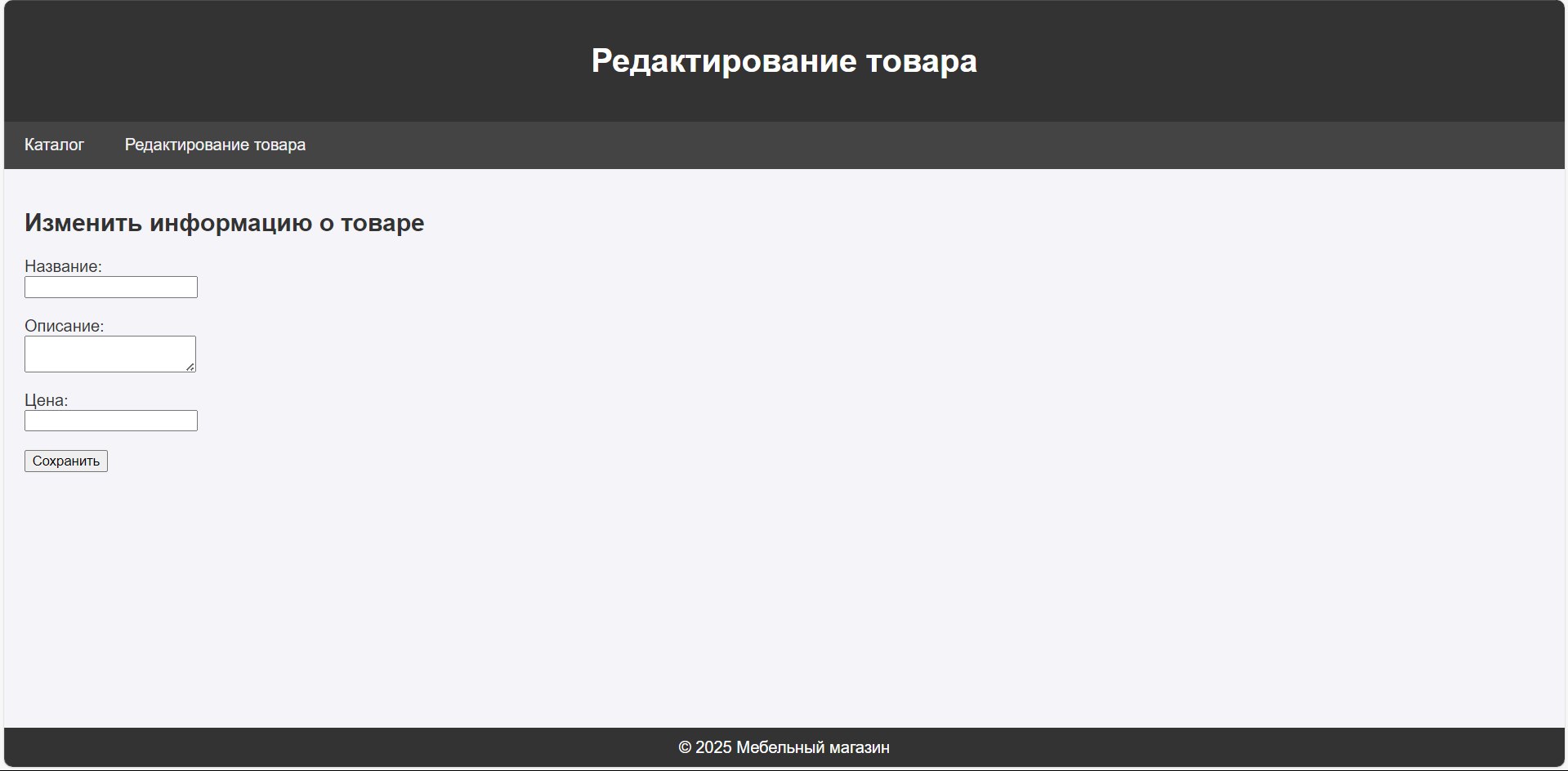


Рисунок – Четвертый пример навигации по сайту.

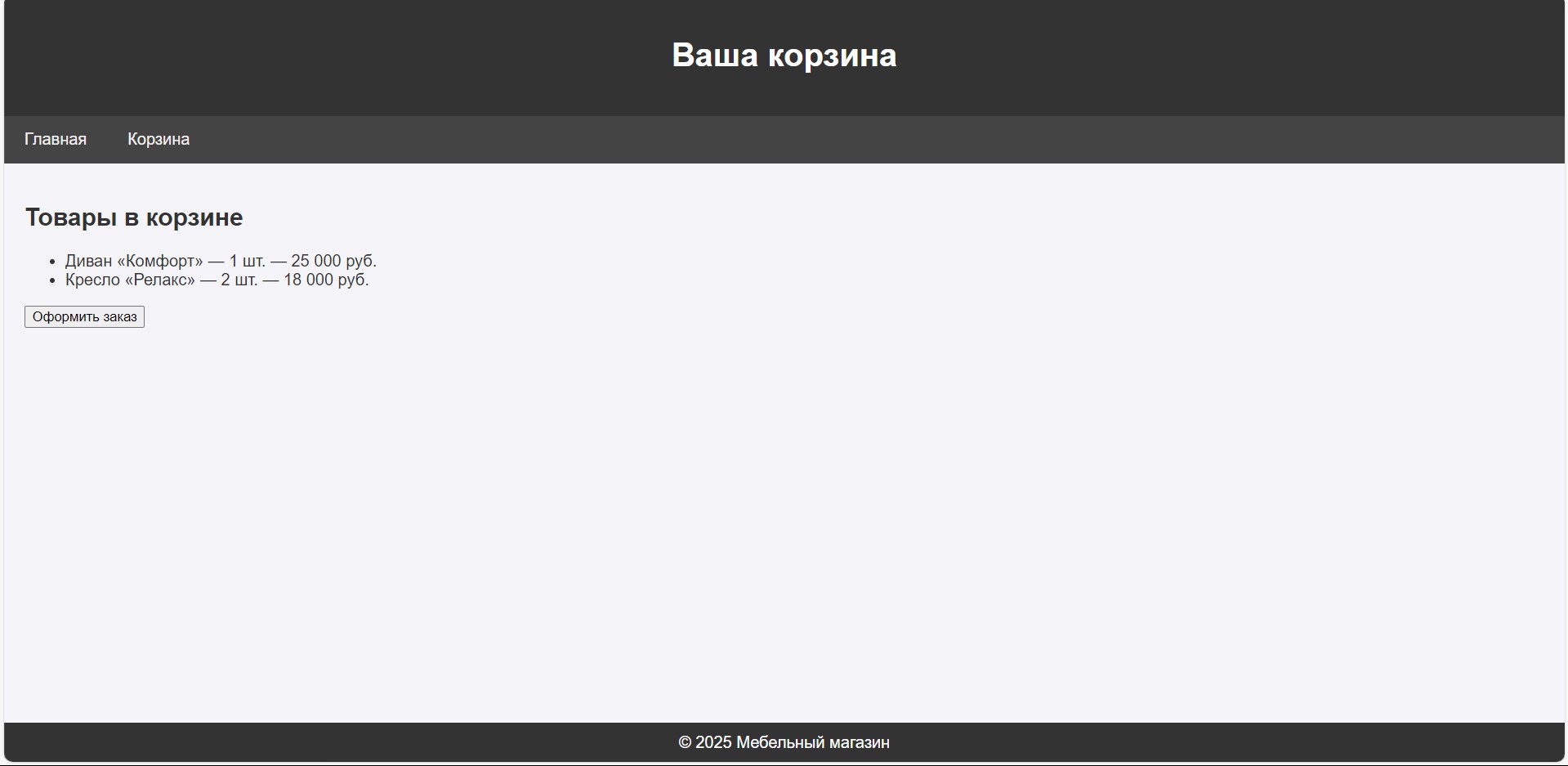


Рисунок – Пятый пример навигации по сайту.

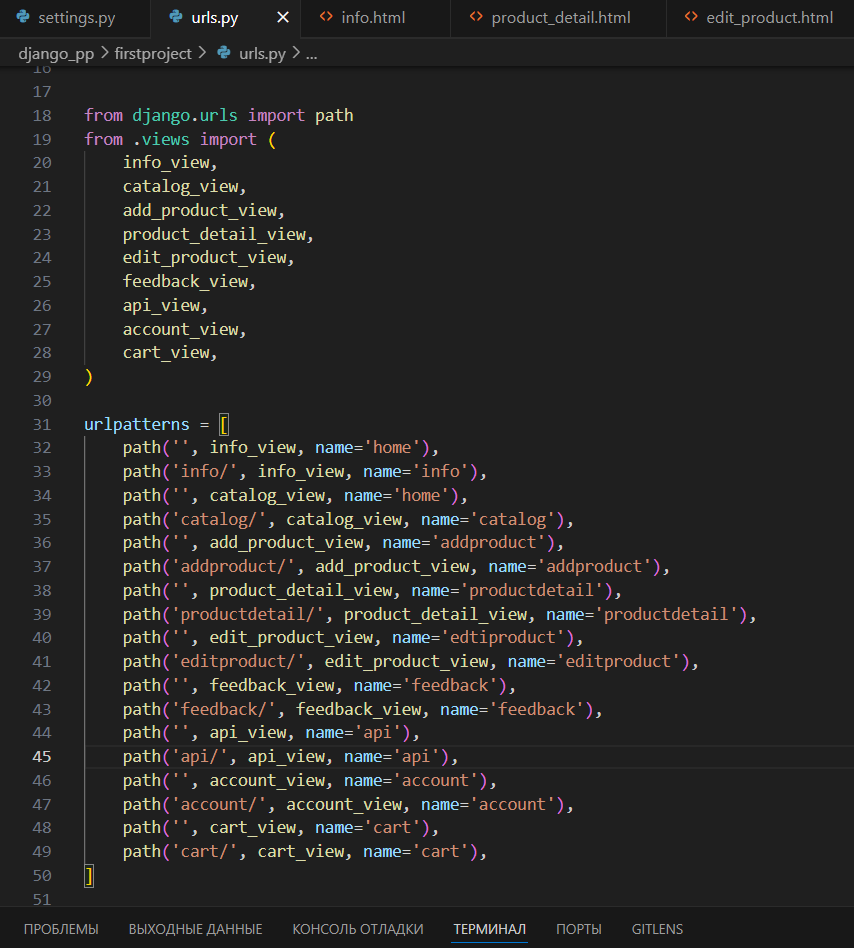


Рисунок – Распределение ссылок для перехода.



Рисунок - Создание представлений.

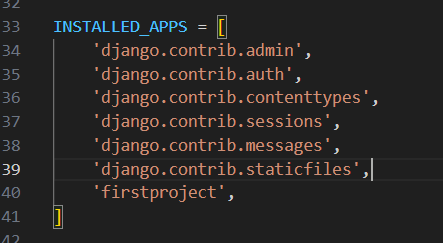


Рисунок – Добавление нового проекта.



Рисунок – Подключение к базе данных.

Вывод: Реализовали структуру маршрутизации в соответствии с прикрепленным шаблоном. В страницах, где предполагается наличие идентификаторов (просмотр товара, изменение товара) или передача других данных реализовали маршрут в соответствии с наличием и применением этих данных. Создали базовые верстки для отображения страниц без полного функционала, как было показано в примере.

# ПРАКТИЧЕСКАЯ РАБОТА №2

Цель работы: Создать структуру базы данных через django и создать формы для добавления.

Ход работы:

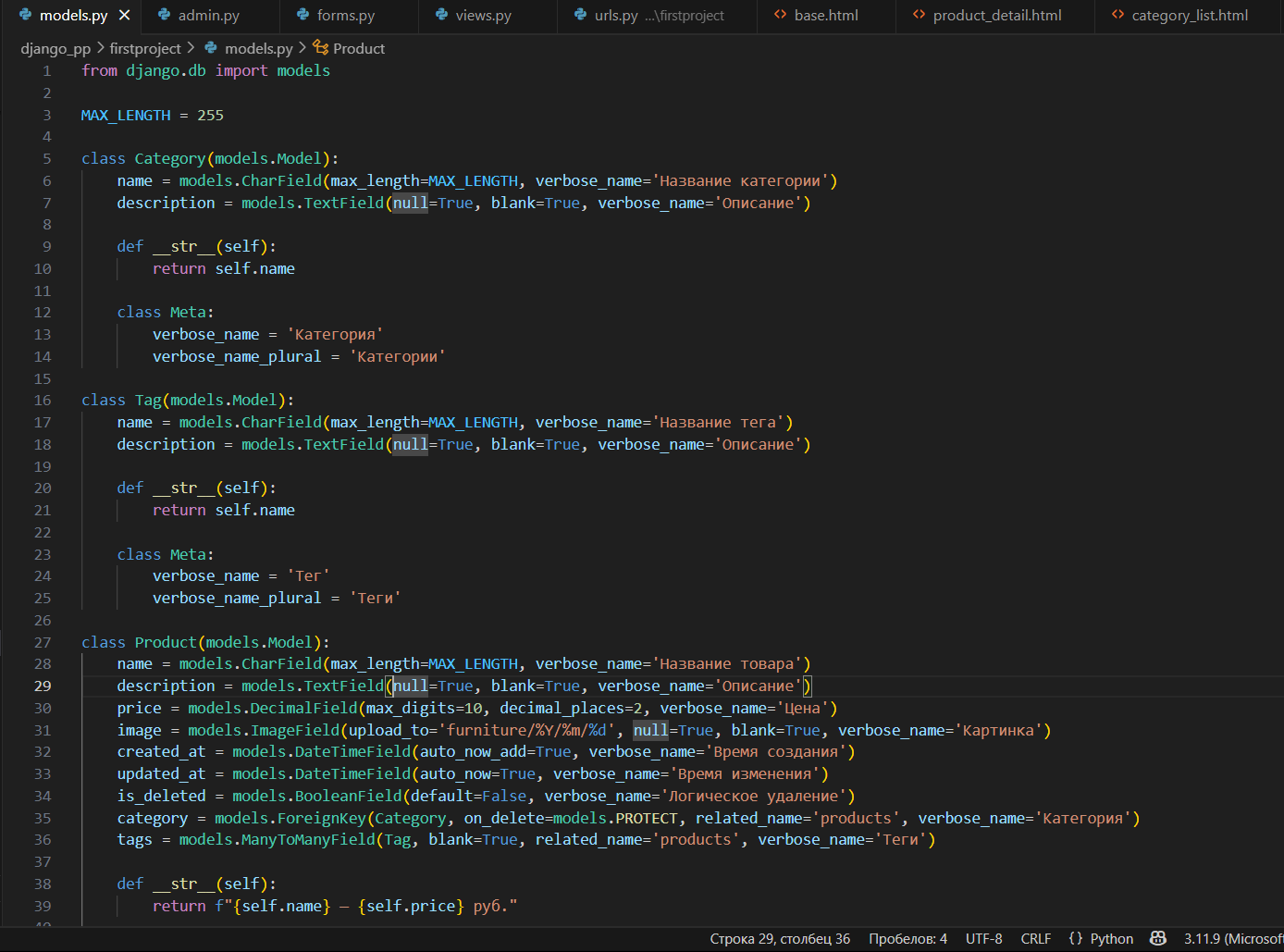


Рисунок – Создание моделей.

В этом файле были созданы модели, а потом уже при помощи команды миграции они были перенесены в базу данных.

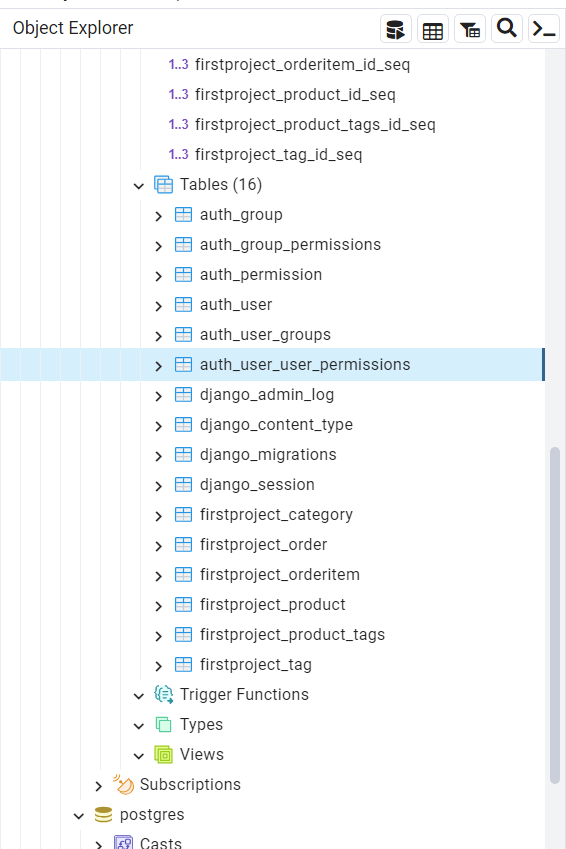


Рисунок – База данных.

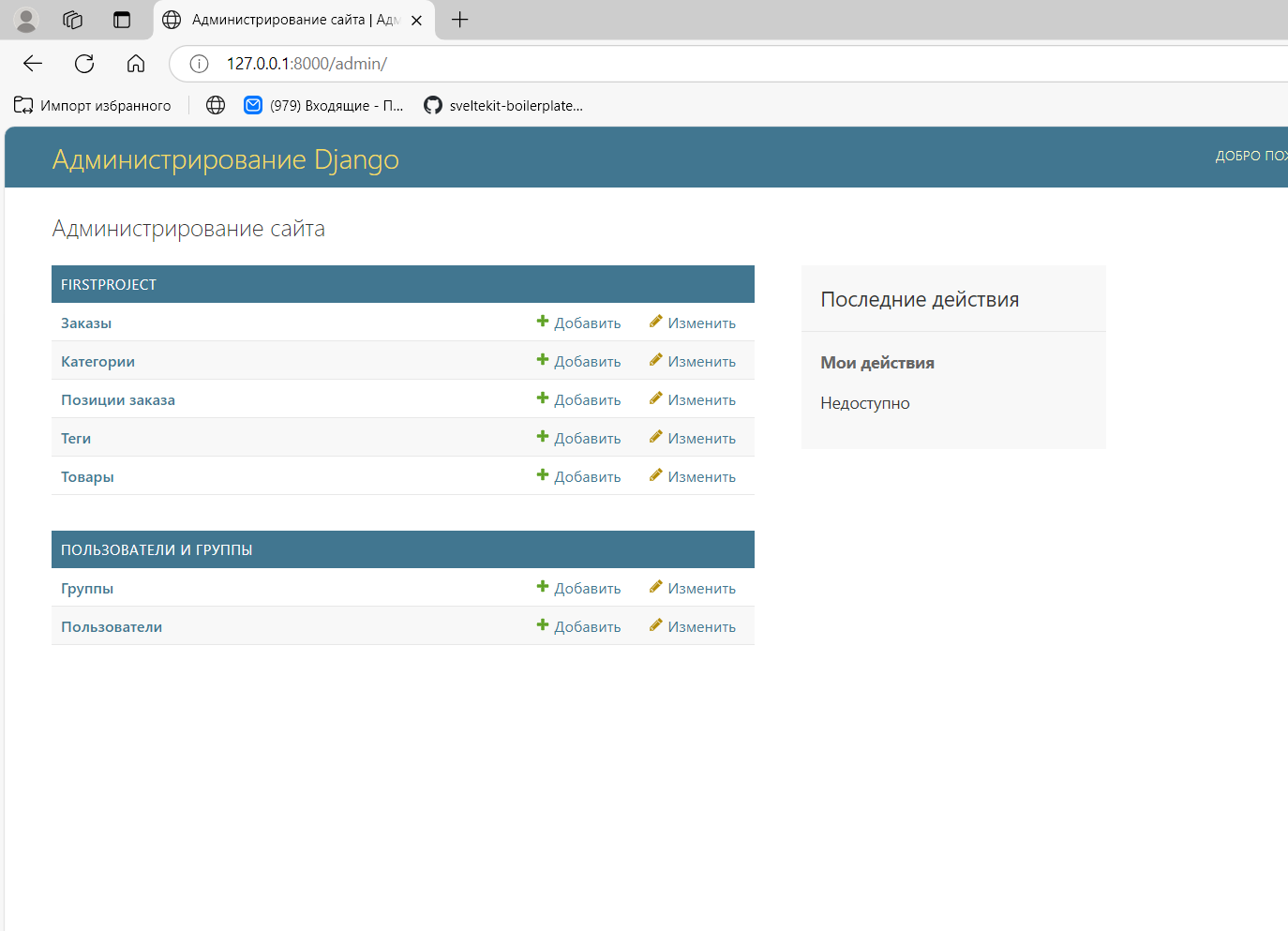


Рисунок – Административная панель.

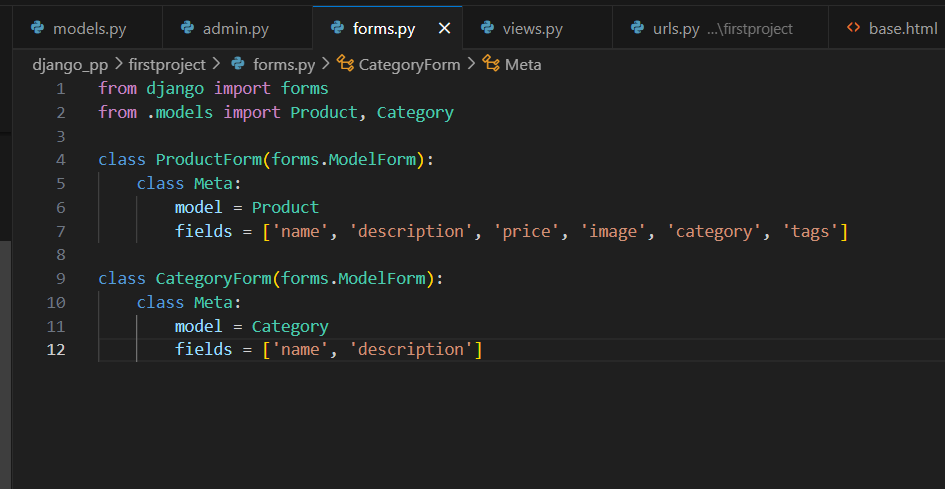


Рисунок – Заполнение форм.

Здесь прописываем то, что у нас будет заполняться.

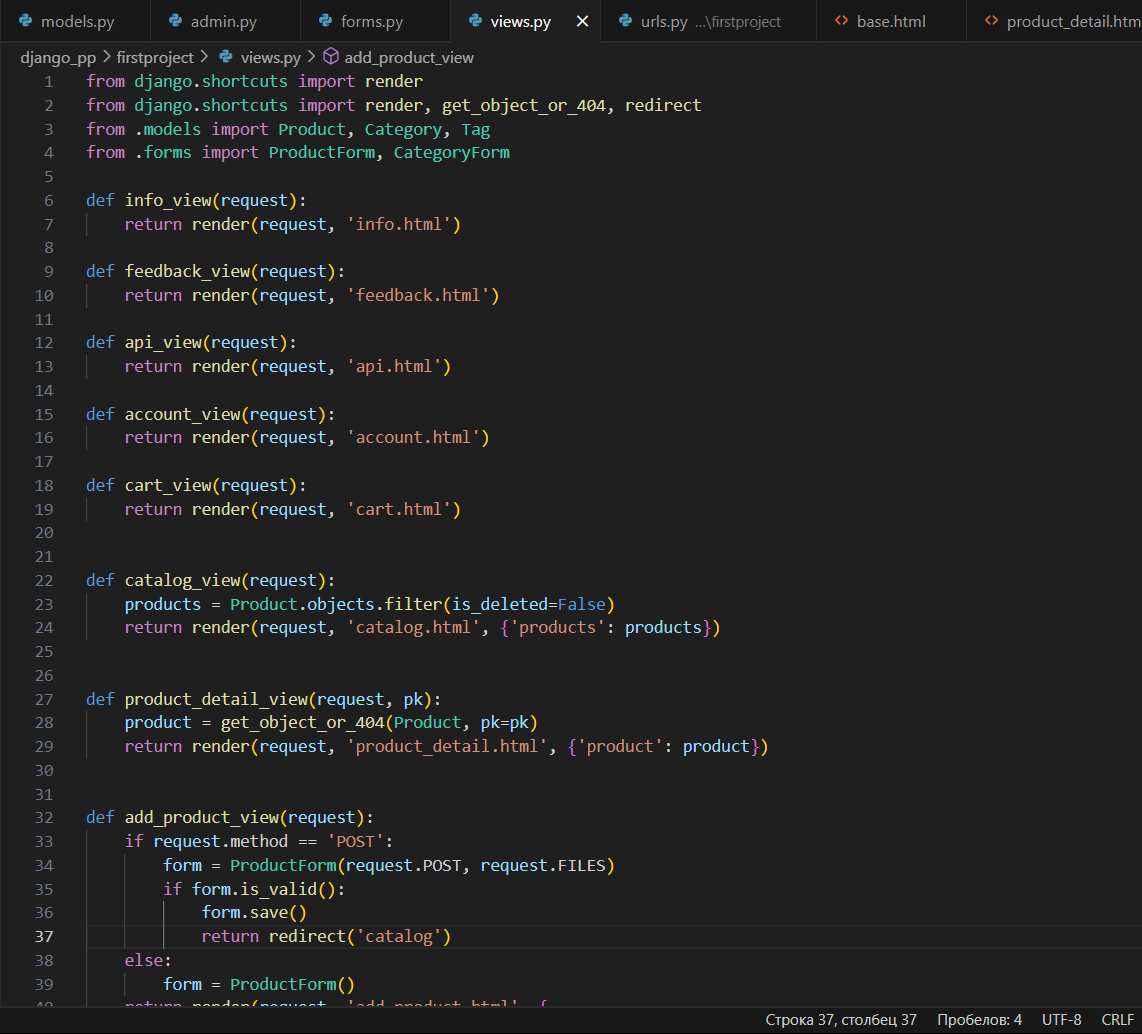


Рисунок – Круд операции.

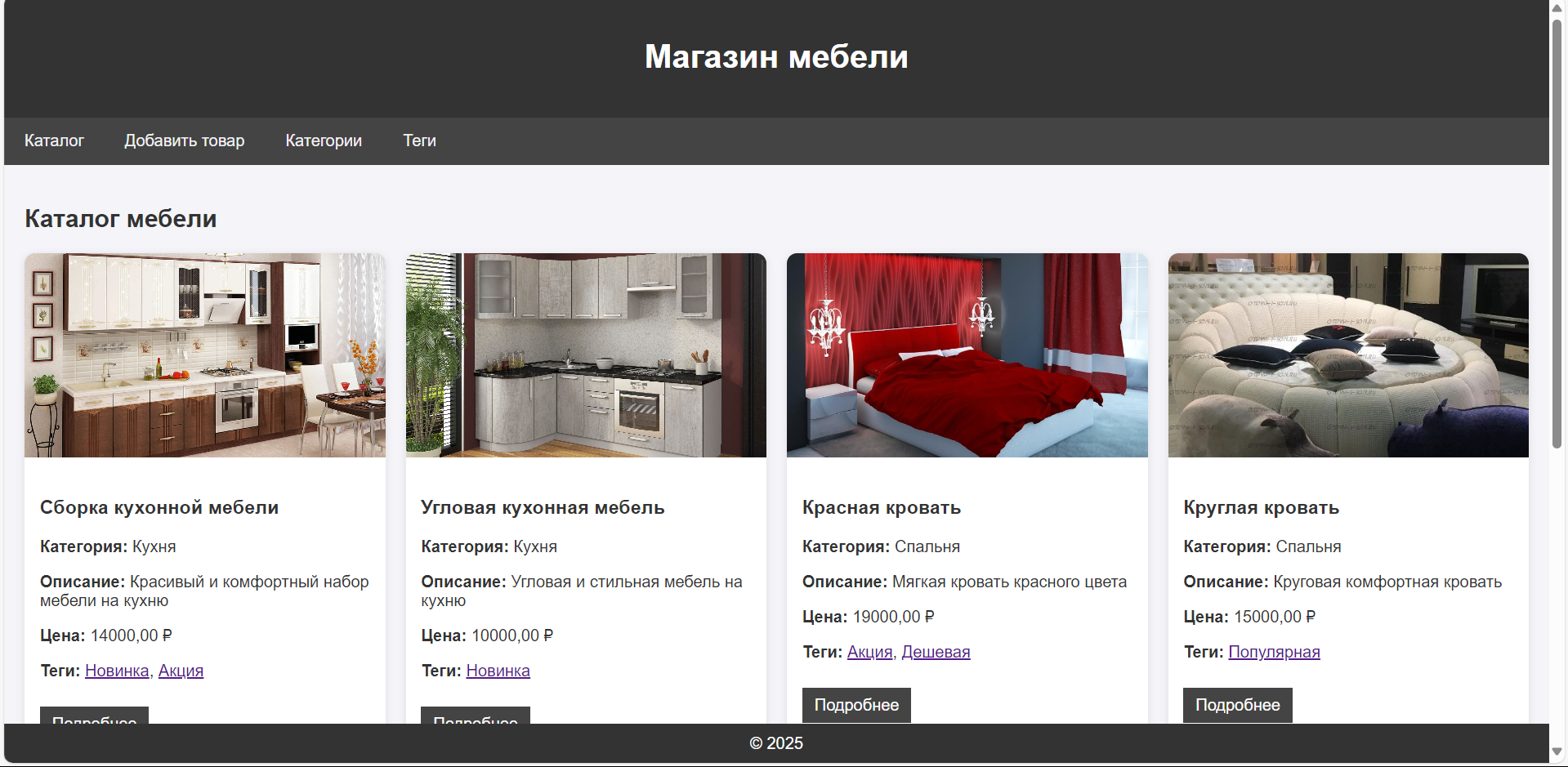


Рисунок – Результат добавления товаров.



Рисунок – Добавление товара.

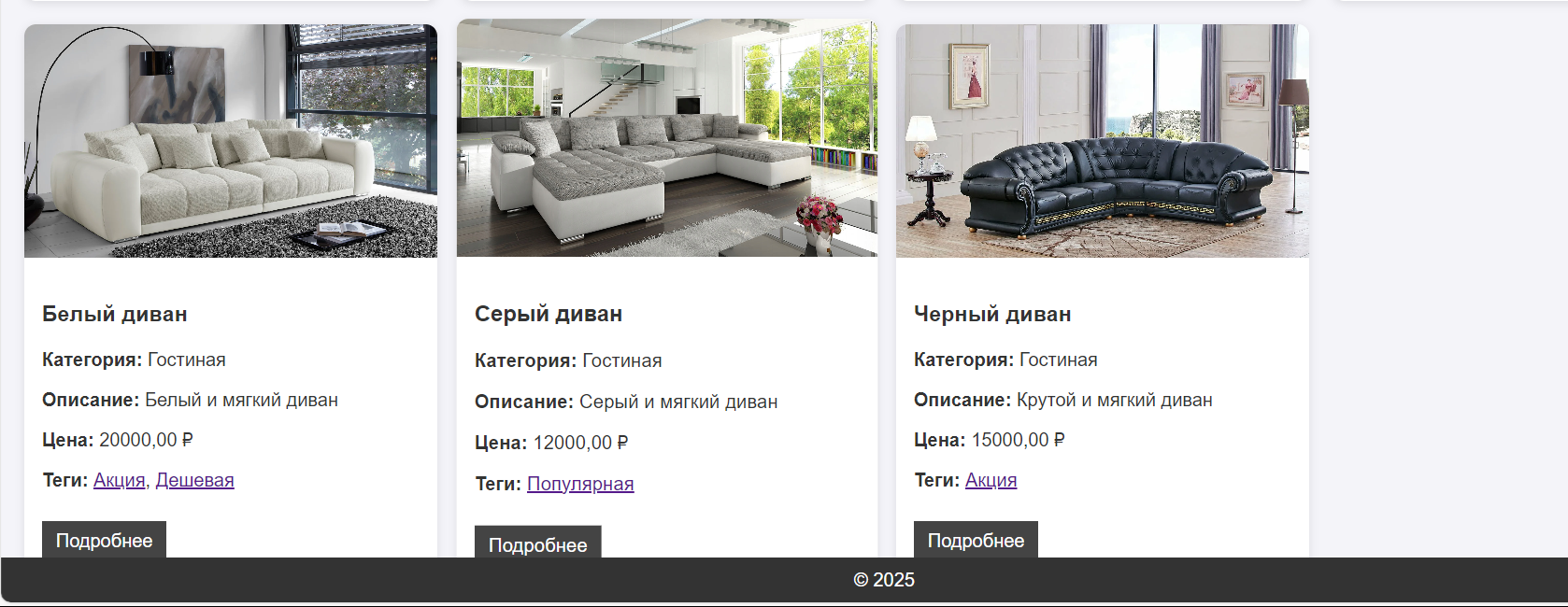


Рисунок – Результат добавления товара.

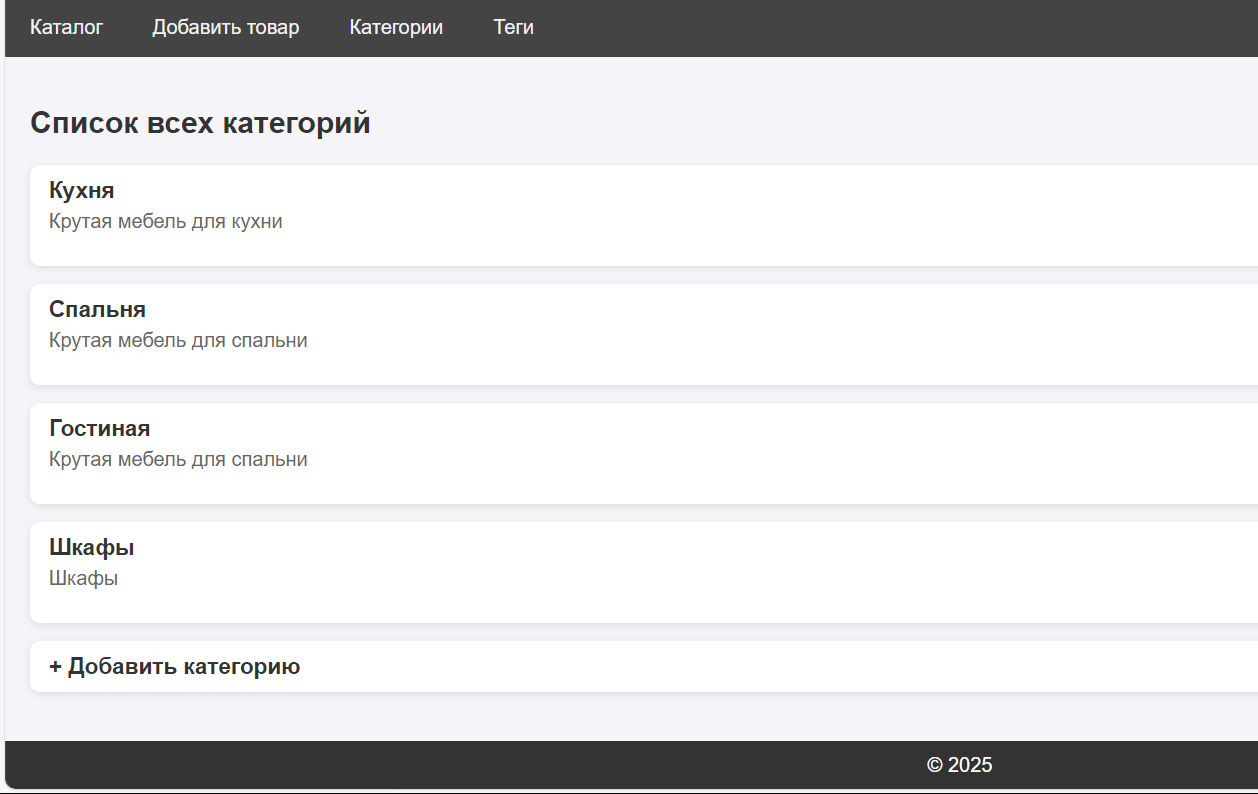


Рисунок – Категории.

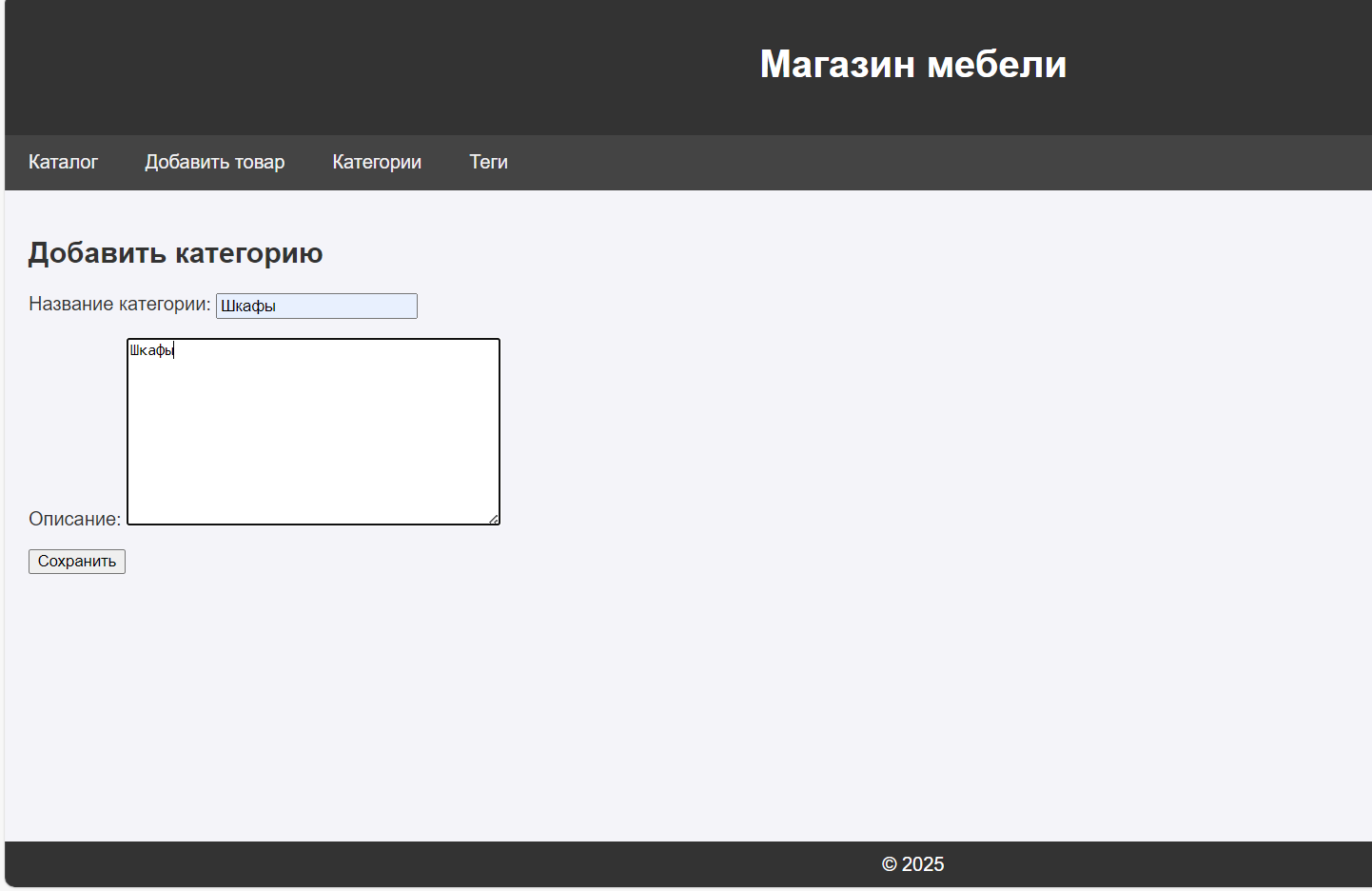


Рисунок – Добавление категории.

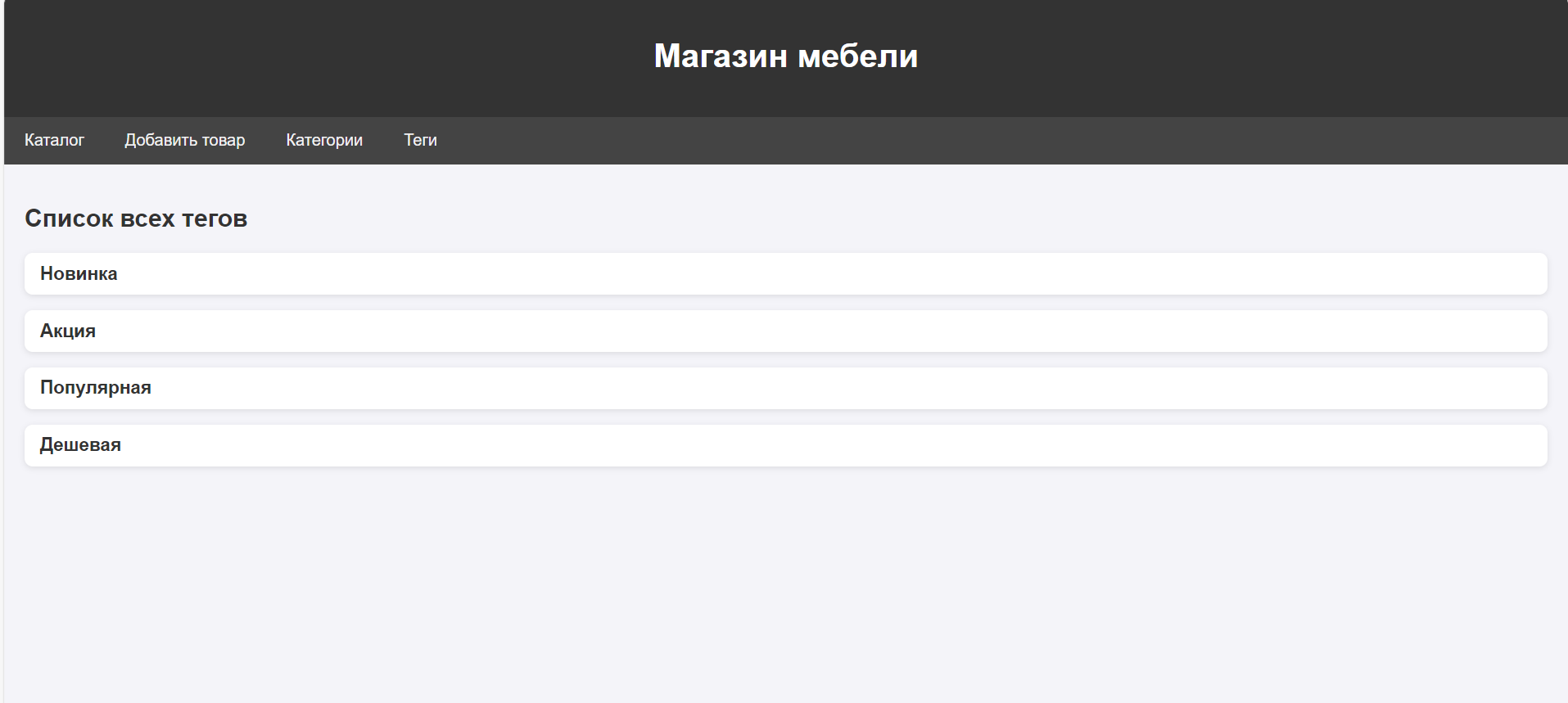


Рисунок – Теги.

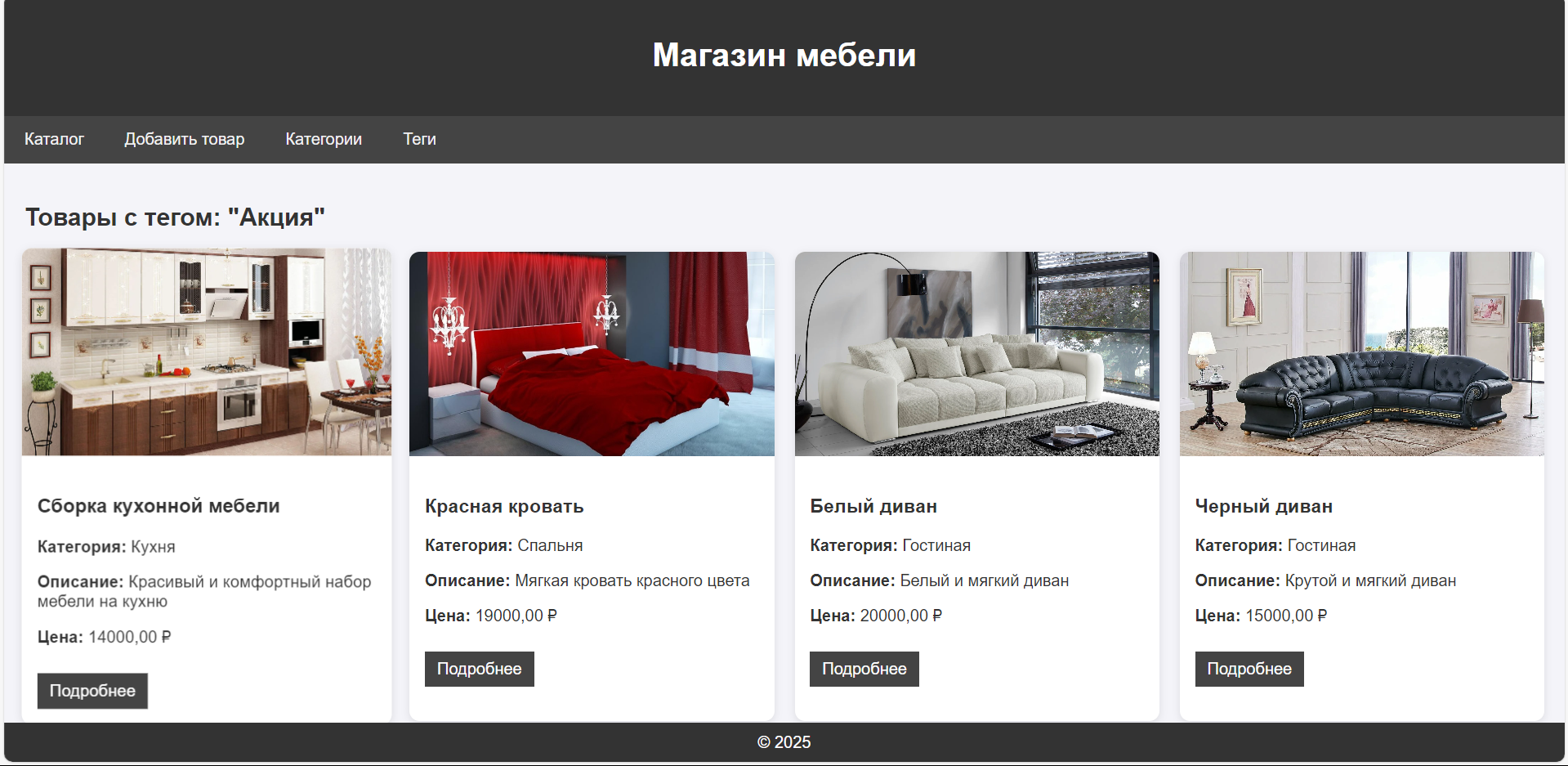


Рисунок – Переход по категории.

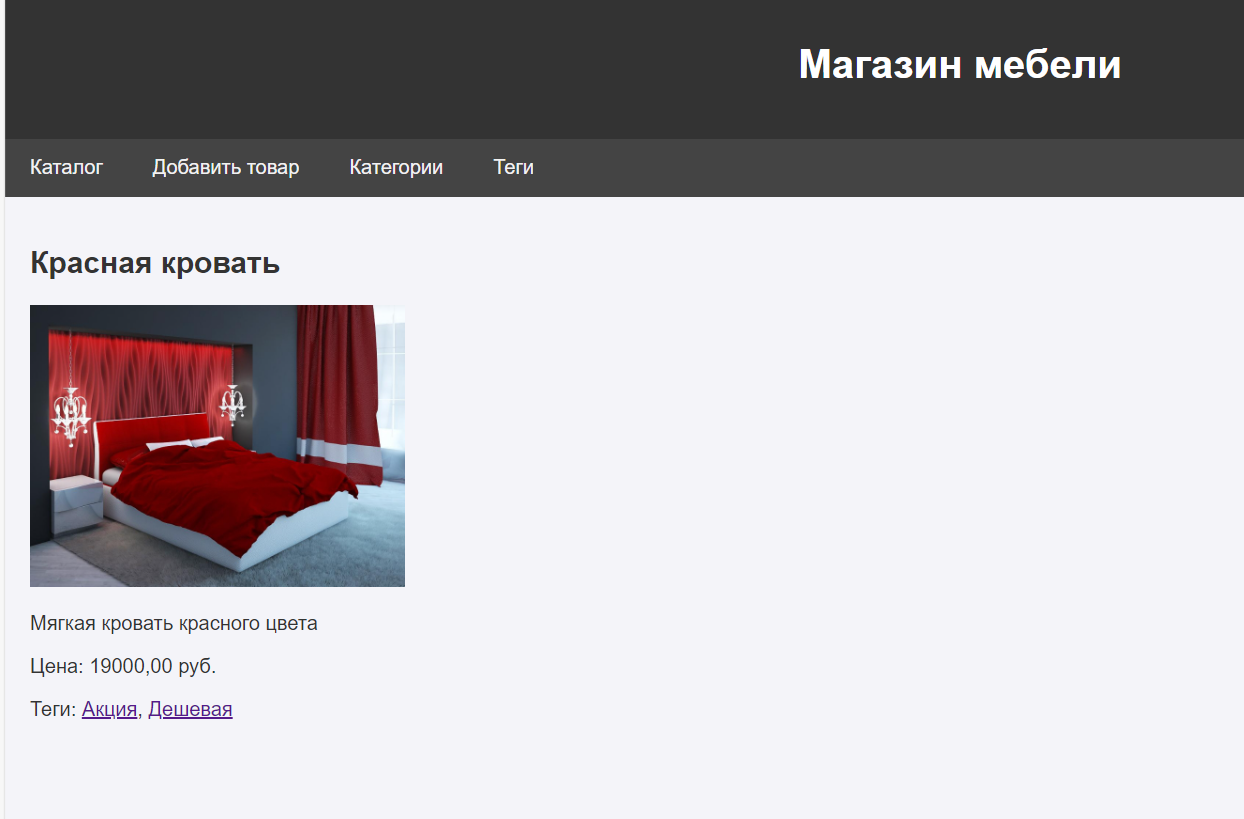


Рисунок – Переход по кнопке "Подробнее".

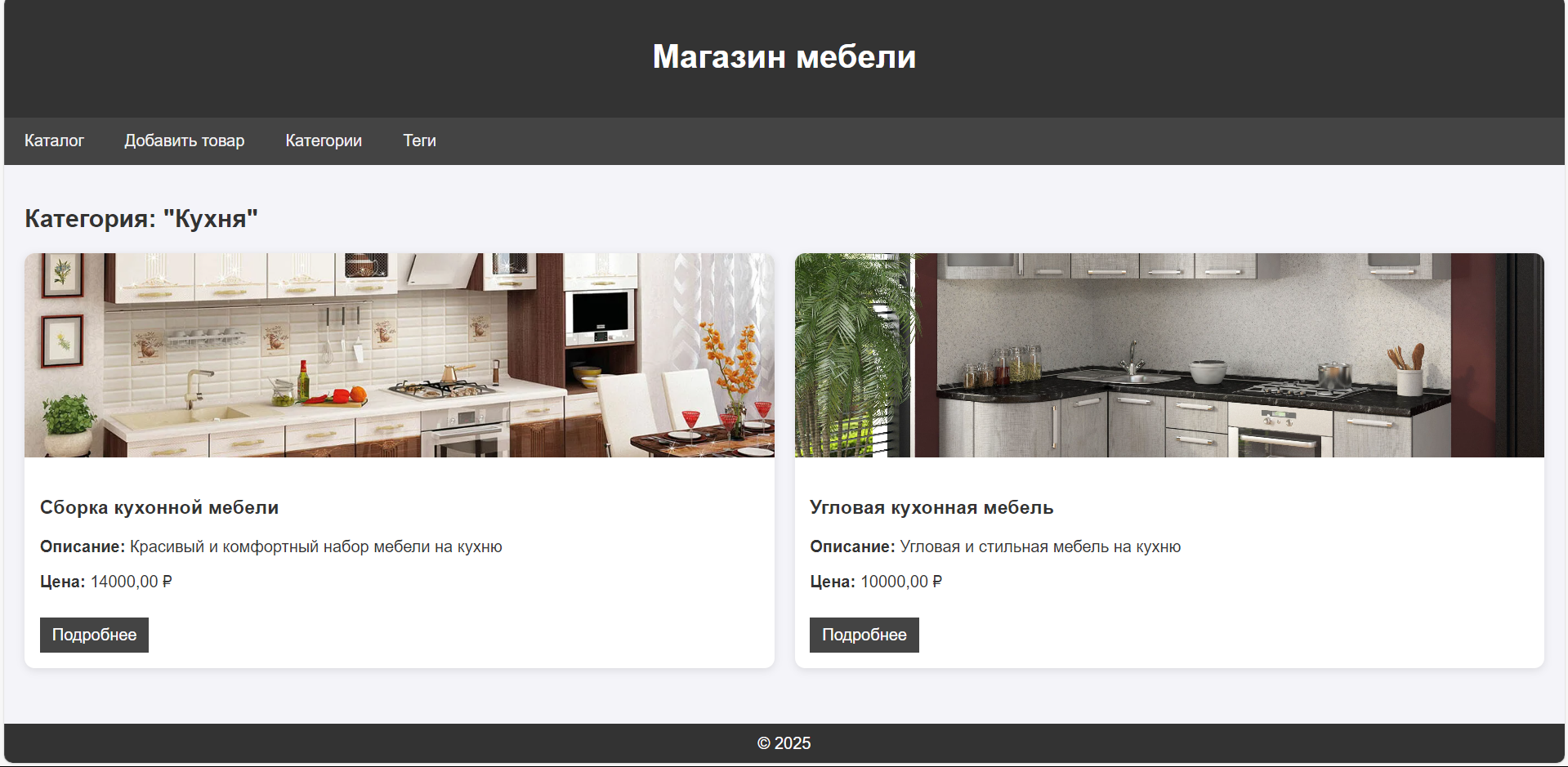


Рисунок – Переход по категориям.

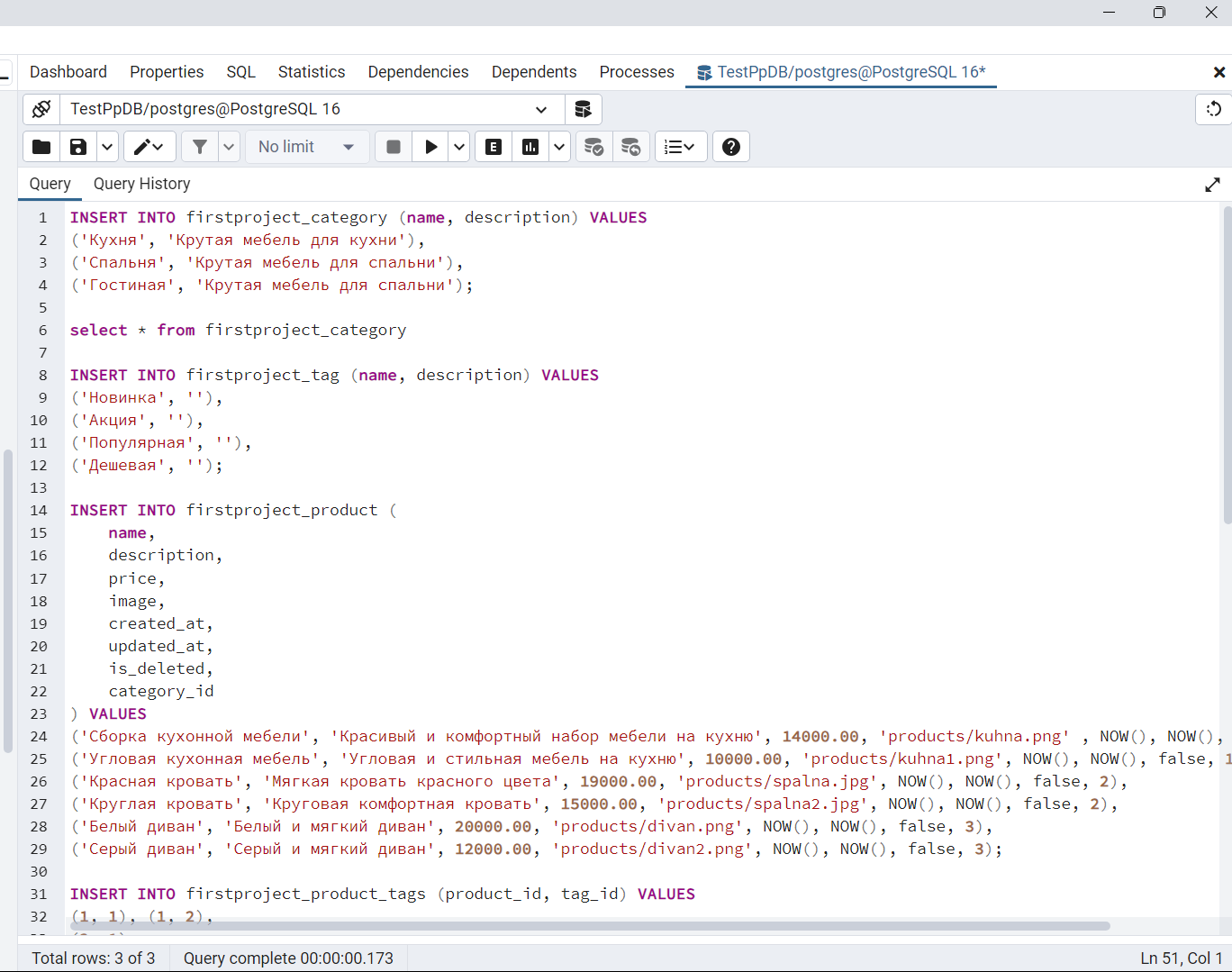


Рисунок – Скрипт базы данных.

Вывод: Создали таблицу "Товар" с атрибутами: название, описание, цена, картинка, время создания, время изменения ,логическое удаление. Создали таблицу "Категория" с атрибутами: Название, описание. Связать с таблицей "Товар" как Один ко Многим. Создали таблицу "Тег" с атрибутами: Название, описание. Связать с таблицей "Товар" Многие ко Многим. Создали таблицу "Заказ" с атрибутами: уникальный номер, дата создание, адрес доставки, телефон клиента, ФИО клиента. Связать с таблицей "Товар". Многие ко многим, с использованием промежуточной таблицы ("Позиция заказа"), где указано "кол-во товаров на позиции" и "скидка за единицу товара". Реализовали вывод товаров на страницу каталога, добавление товара (с использованием форм) и вывод одного товара на страницу. При выводе одного товара выводится список тегов, при нажатии на которых, происходит переход на список товаров, связанных с этим тегом. Реализовали страницу списка категорий, при нажатии на категорию, происходит переход на страницу вывода товаров с данной категорией. Реализовали страницу добавления категории с применением форм. Реализовали страницу вывода тегов (см. выше) и страницу списка существующих тегов. Заполнили базу данных: товар - минимум 6 записей; категория - минимум 2 записи; тег - минимум 4 записи; заказ - минимум 3 записи; позиция заказа минимум 8 записи.

# ПРАКТИЧЕСКАЯ РАБОТА №3

Цель работы: Ознакомиться с работой шаблонов и реализовать наследование шаблонов.

Ход работы:

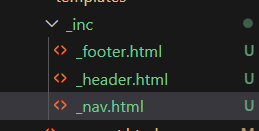


Рисунок – Добавление компонентов.



Рисунок – Скелет.

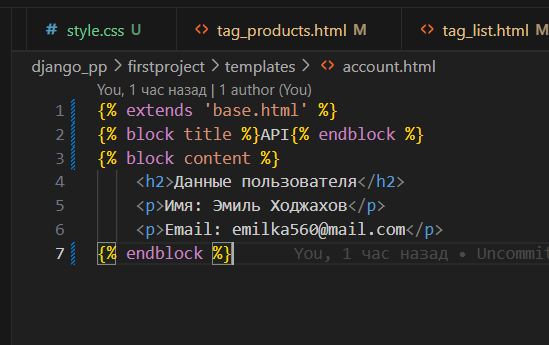


Рисунок – Пример измененной страницы.

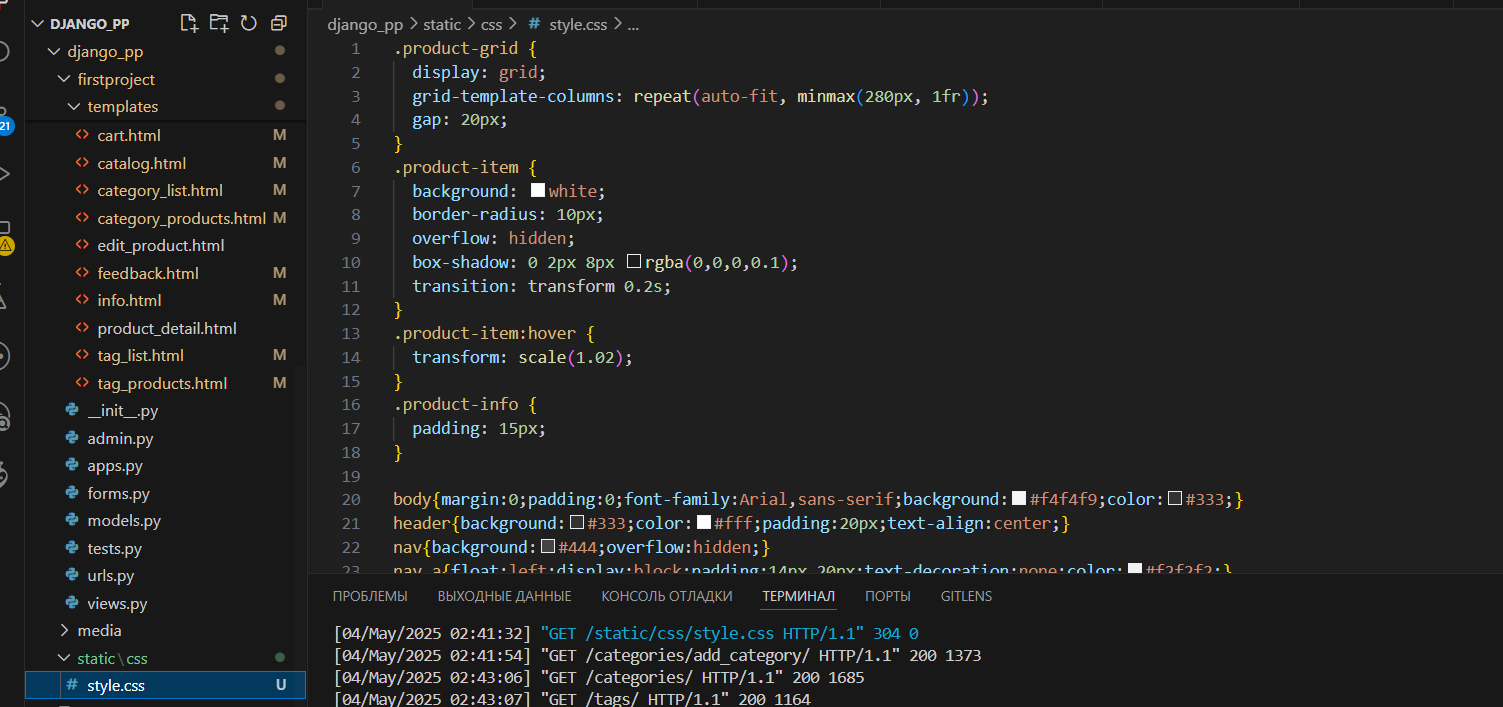


Рисунок – Стили.

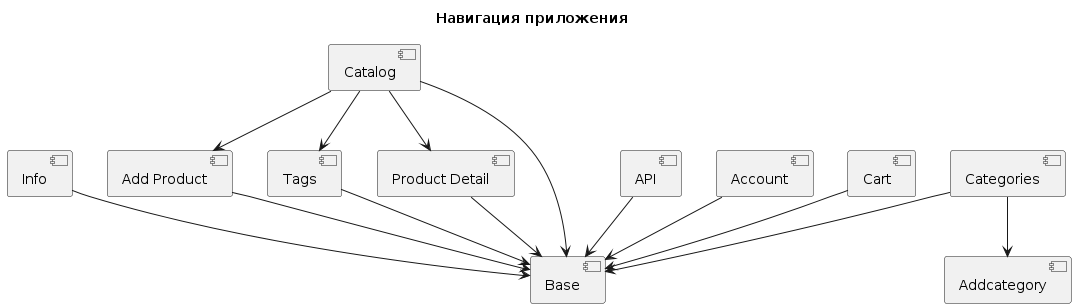


Рисунок – Диаграмма.

Вывод: Реализовали наследование шаблонов, со структурой (base.html), компонентами(\_header.html, \_footer.html) и страницами наполнения, зарисовали диаграмму компонентов, в скелете привели необходимые элементы и применили следующие компоненты (header, footer, nav).

# ПРАКТИЧЕСКАЯ РАБОТА №4

Цель работы: Написать функционал взаимодействия с сессиями. Добавить на свой сайт вкладку корзины.

Ход работы:

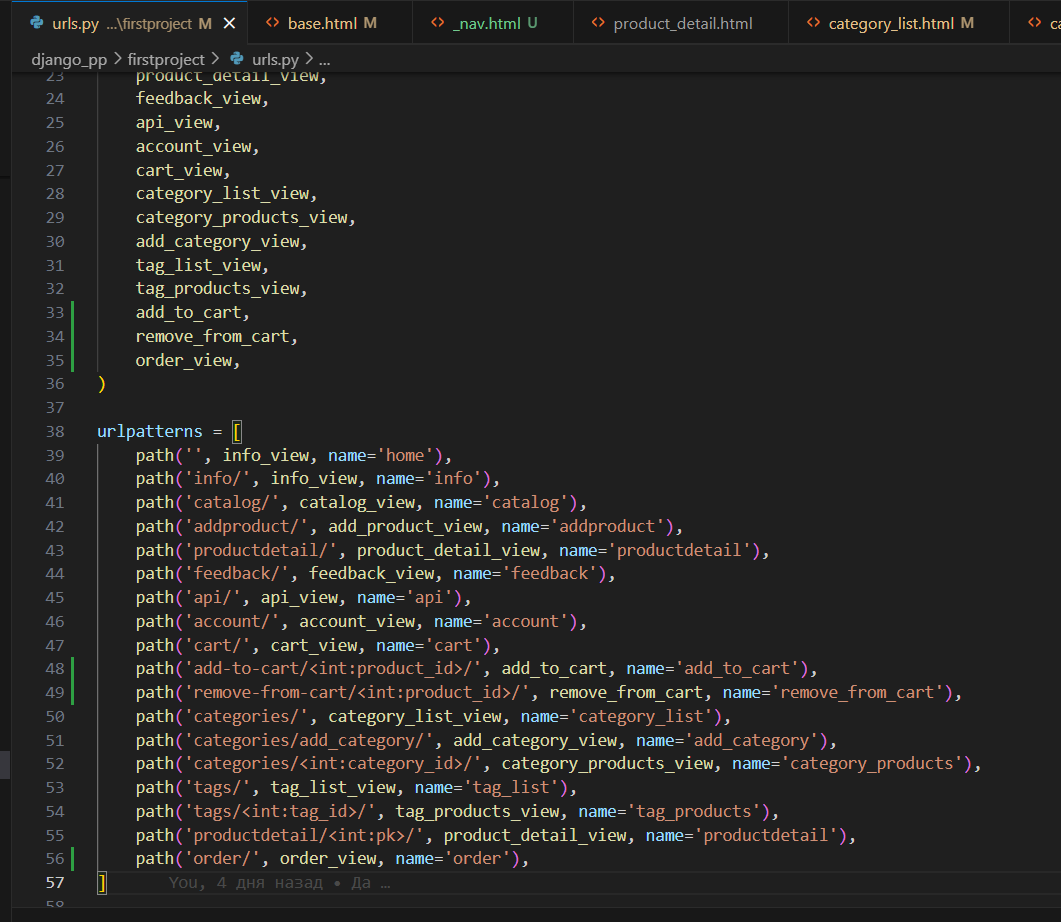


Рисунок – Добавление ссылок.

Добавили сюда 3 ссылки: add\_to\_cart, remove\_from\_cart и order\_view.

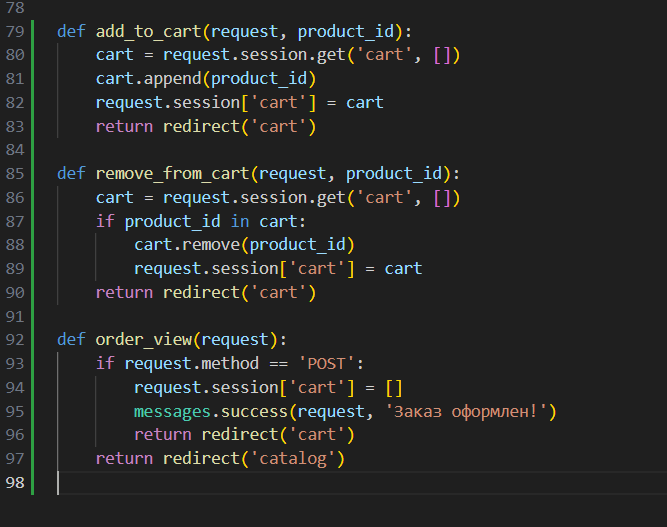


Рисунок – Добавление новых представлений.

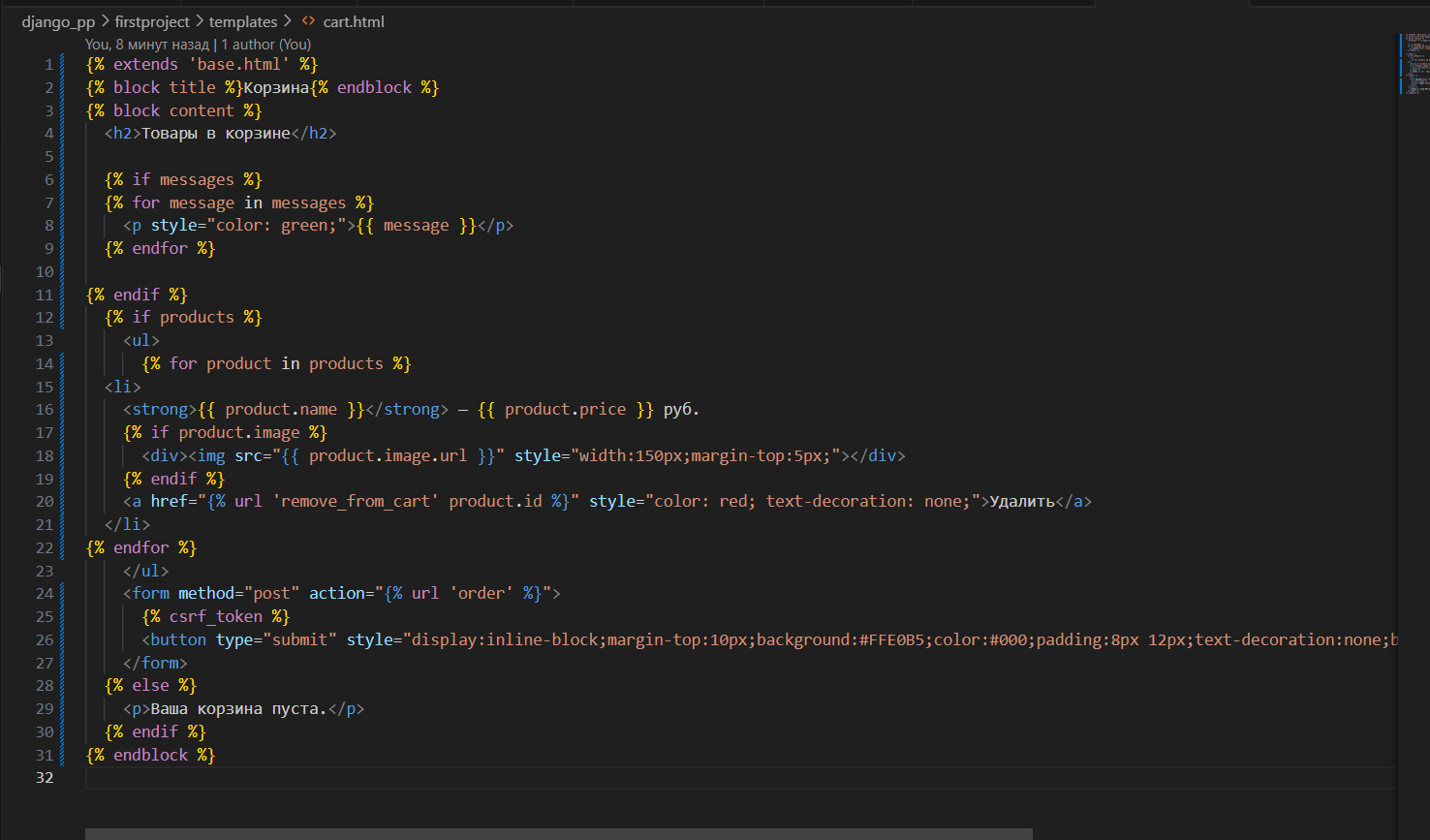


Рисунок – Корзина.

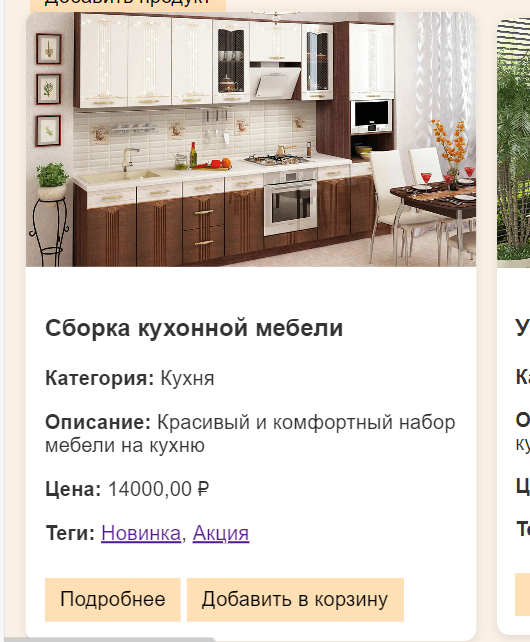


Рисунок – Результат.

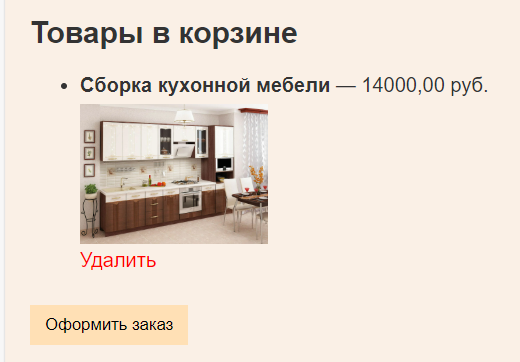


Рисунок – Результат добавления в корзину.



Рисунок – Результат удаления с корзины.



Рисунок – Результат оформления заказа.



Рисунок - Результат нескольких товаров в корзине.

Вывод: Написали функционал взаимодействия с сессиями. Добавили на свой сайт вкладку корзины.

# ПРАКТИЧЕСКАЯ РАБОТА №5

Цель работы: Создание API и разграничение при помощи прав доступа. Реализовать API в вашем проекте, в котором должны быть все CRUD операции (для всех таблиц). Реализовать в классах ListView, DetailView, CreateView, UpdateView, DeleteView нужные права доступа при помощи @permission\_required(). Реализовать @login\_required на всех страницах, где это необходимо, чтобы неавторизированный пользователь не мог на них прейти. Создать 10 различных пользователей, которые разделяются на (права доступа желательно раздавать при помощи "Групп"); Суперпользователь (1 пользователь); Продавец (3 пользователя) - имеют право на добавление, изменение и удаление (удаление происходит логическое) необходимых товаров и возврат товаров (если они логически удалены); Администратор (2 пользователя) - имеют право на добавление, изменение и удаление (физическое и логическое) необходимых товаров, тегов и категорий Покупатель (4 пользователя) - имеют право на просмотр списка товаров и его подробной информации, перемещаться по тегам и категориям.

Ход работы:

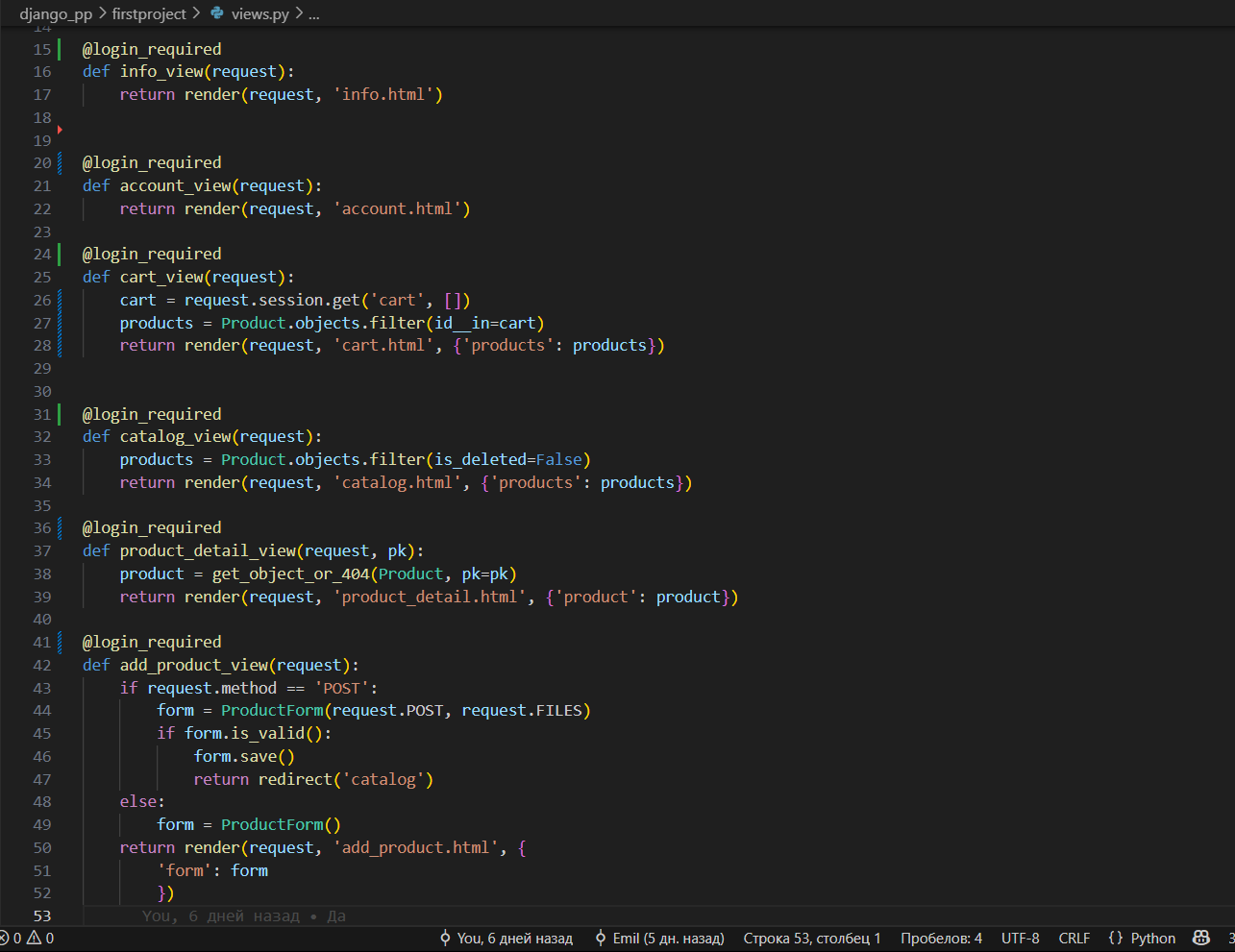


Рисунок – Права доступа для авторизированных и неавторизированных пользователей.

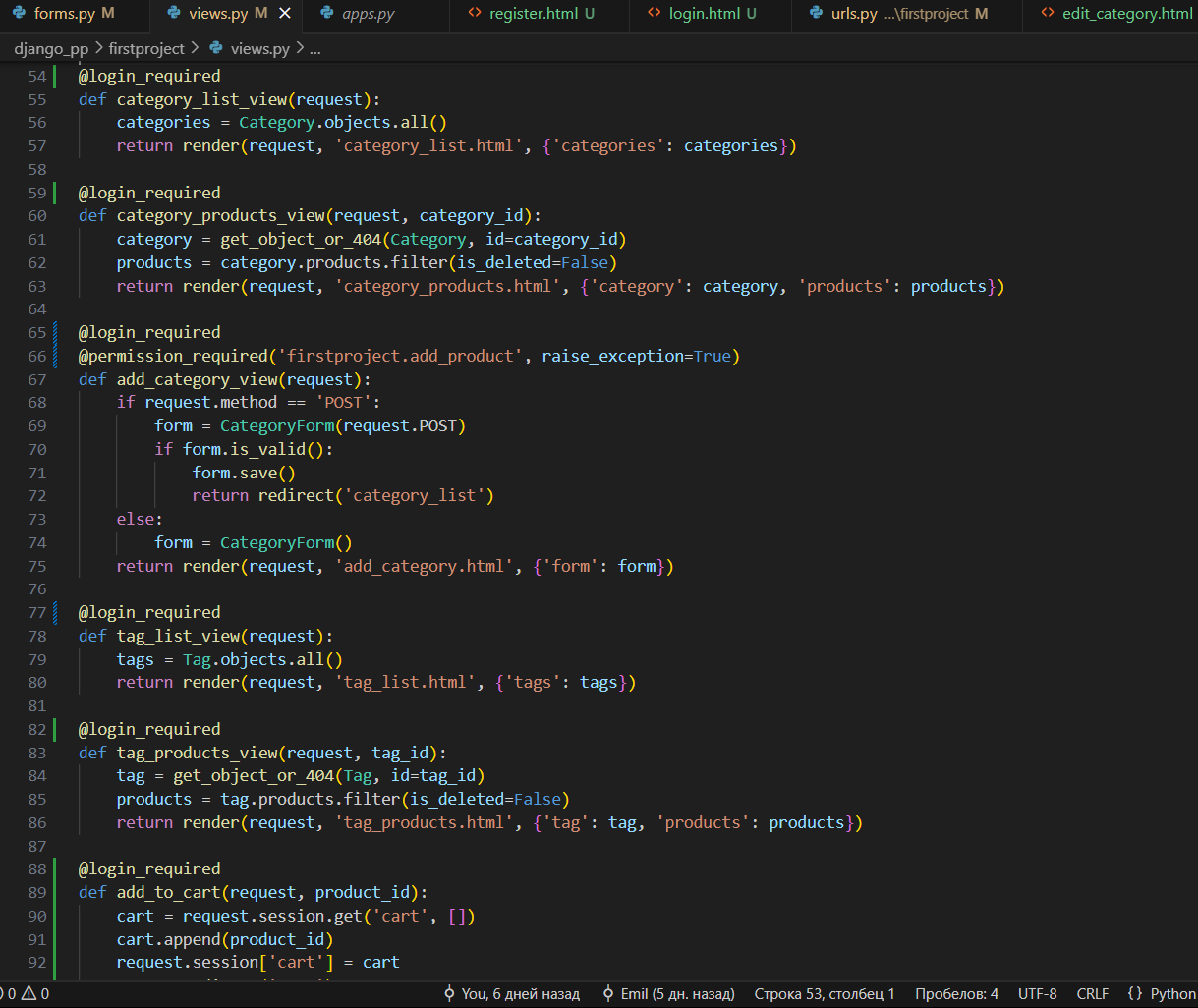


Рисунок – Права доступа.

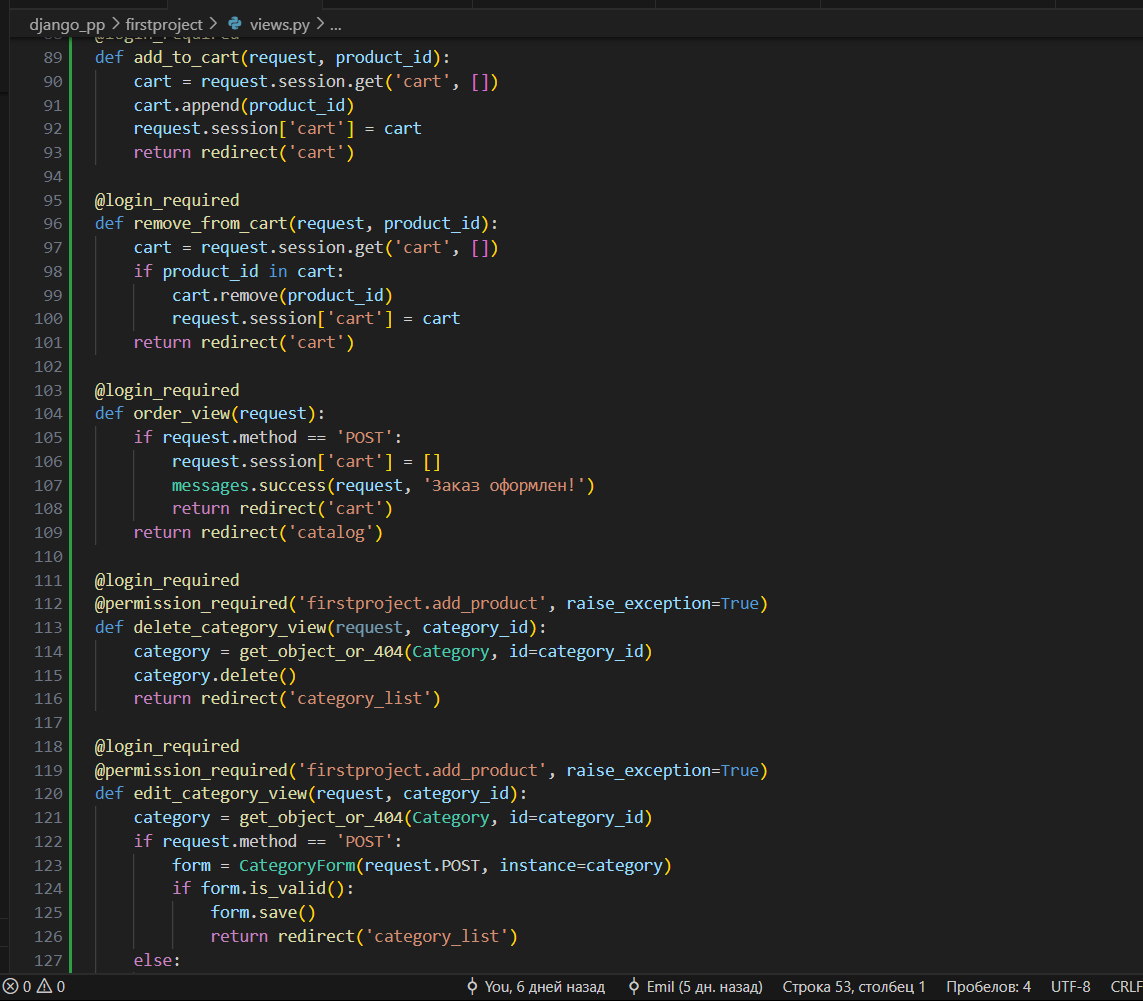


Рисунок – Права доступа.

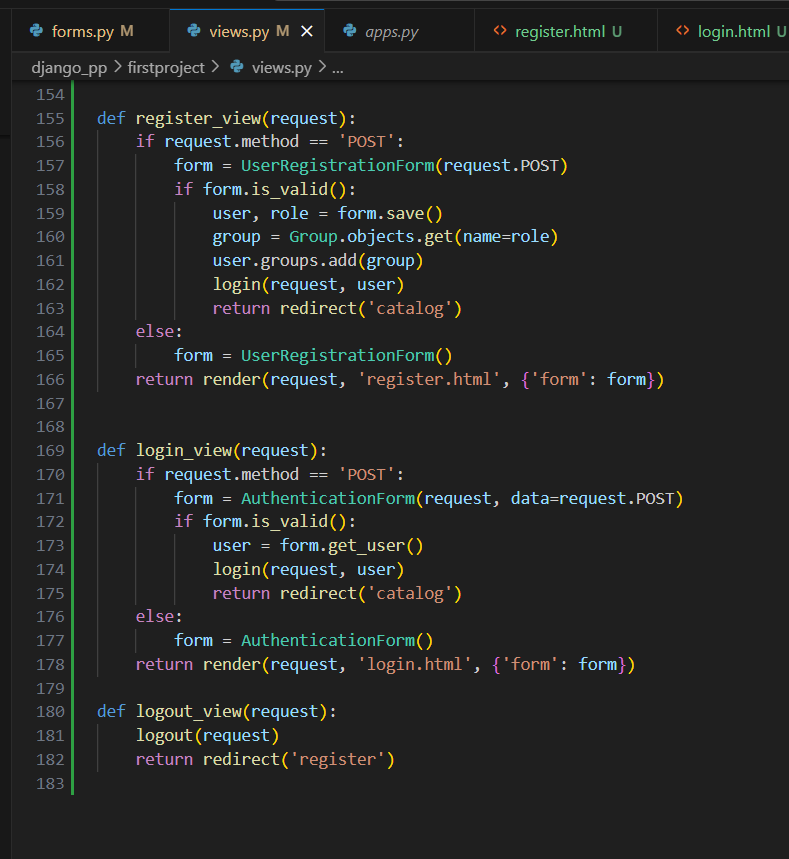


Рисунок – Права доступа.

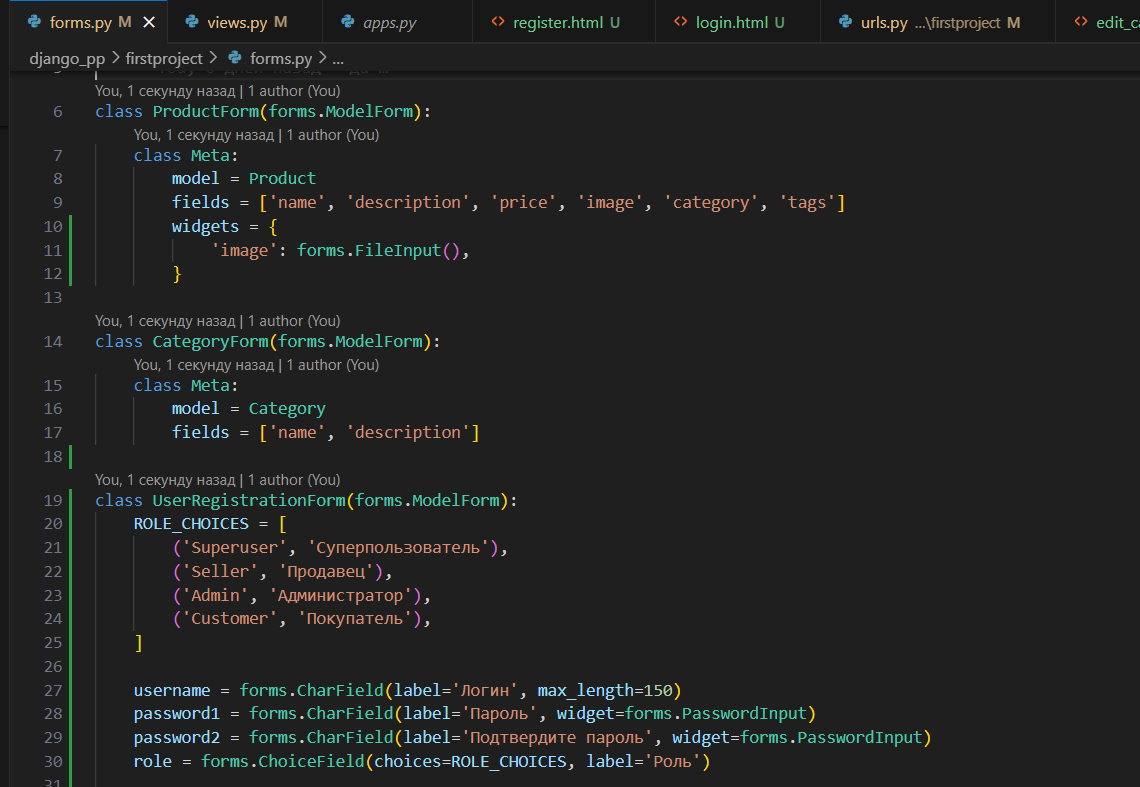


Рисунок – Распределение ролей.

Регистрация пользователя происходит с выбором роли и когда он выбирает роль, то он сразу добавляется в группу с правами, когда же юзер пытается делать что-то выше его прав, то его перекидывает на страницу с ошибкой.

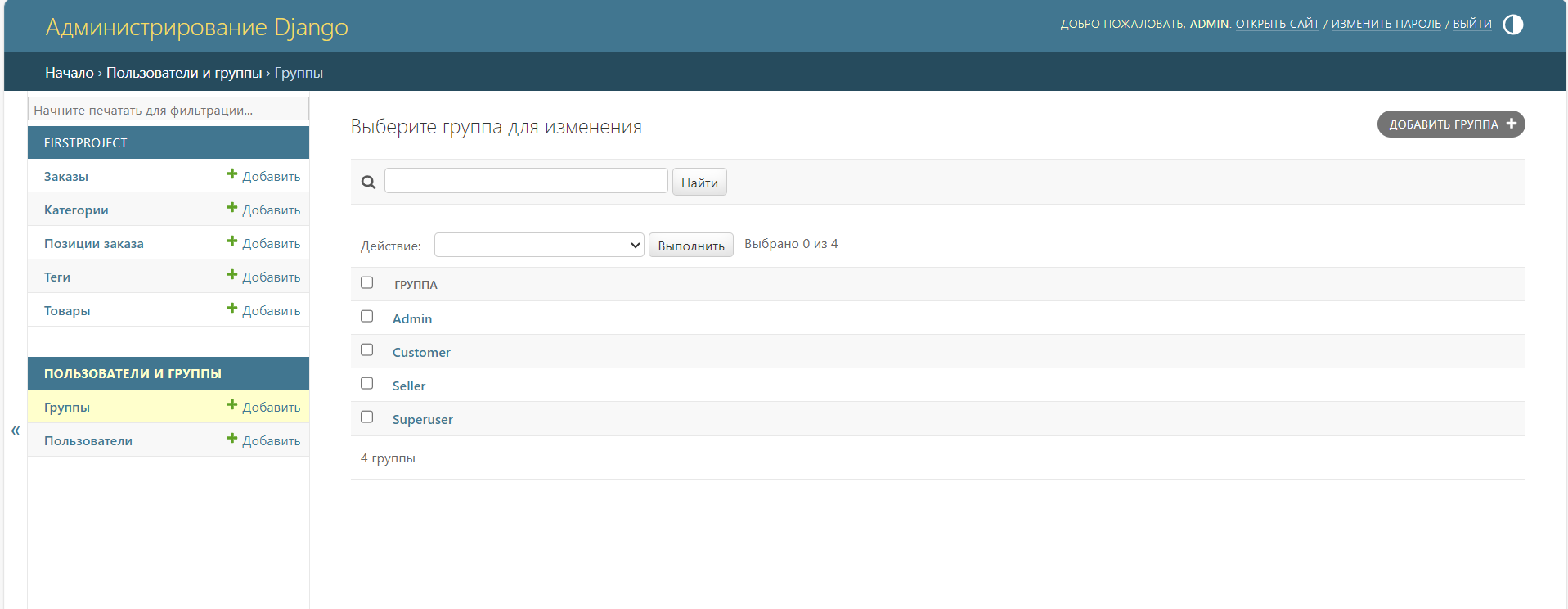


Рисунок – Создание 4 групп.

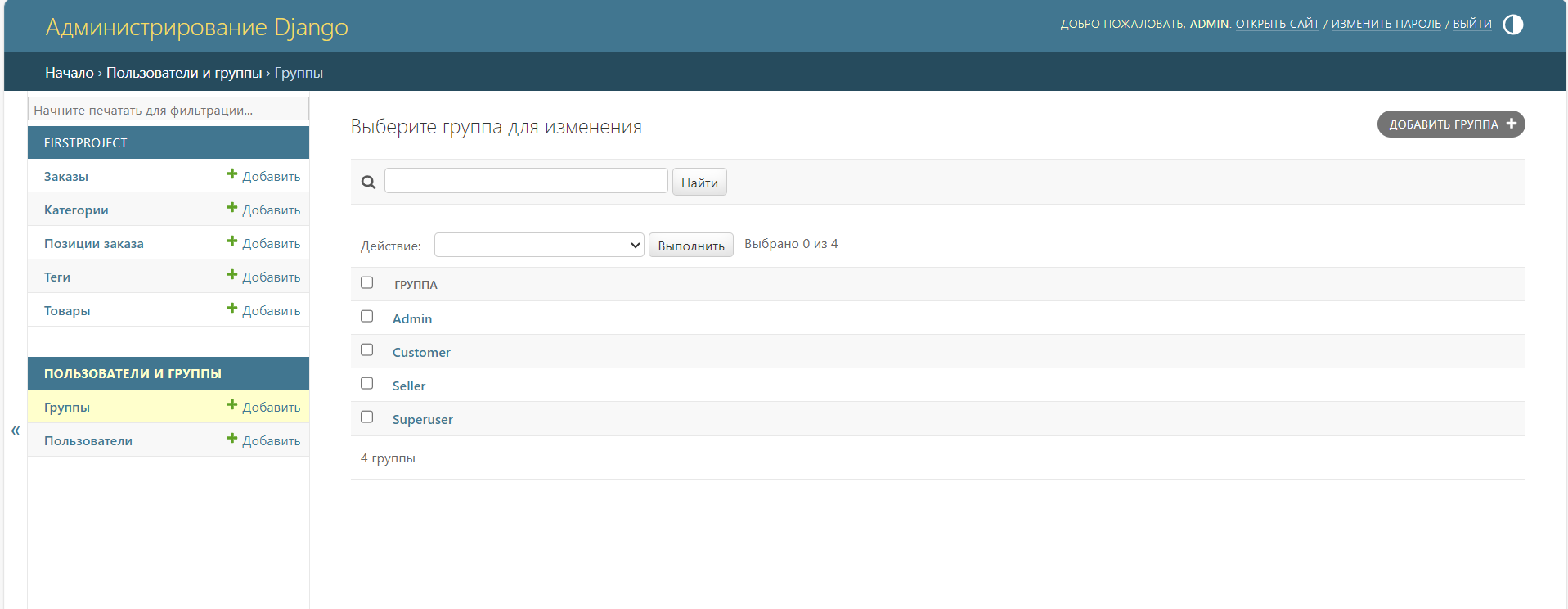


Рисунок – Пример выдачи прав каждой группе.

Вывод: Создали API и разграничили при помощи прав доступа. Р API в вашем проекте, в котором должны быть все CRUD операции (для всех таблиц). Реализовали в классах ListView, DetailView, CreateView, UpdateView, DeleteView нужные права доступа при помощи @permission\_required(). Реализовали @login\_required на всех страницах, где это необходимо, чтобы неавторизированный пользователь не мог на них прейти. Создали 10 различных пользователей, которые разделяются на (права доступа желательно раздавать при помощи "Групп"); Суперпользователь (1 пользователь); Продавец (3 пользователя) - имеют право на добавление, изменение и удаление (удаление происходит логическое) необходимых товаров и возврат товаров (если они логически удалены); Администратор (2 пользователя) - имеют право на добавление, изменение и удаление (физическое и логическое) необходимых товаров, тегов и категорий Покупатель (4 пользователя) - имеют право на просмотр списка товаров и его подробной информации, перемещаться по тегам и категориям.