|  |
| --- |
| Министерство науки и высшего образования РФ |
| Федеральное государственное автономное |
| образовательное учреждение высшего образования |
| **«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»** |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

|  |
| --- |
|  |
| **КУРСОВОЙ ПРОЕКТ** |
| Разработка веб-сайта с каталогом для компании «Крылатые Качели» на Django и React |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| Преподаватель | |  |  |  | П.В. Пересунько |
|  |  |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ23-17/2б, 032322546 |  |  |  | Е.А. Гуртякин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc166503246)

[1 Проектирование 4](#_Toc166503247)

[2 Разработка минимального стоящего продукта 4](#_Toc166503248)

[2.1 Первый спринт 4](#_Toc166503249)

[2.2 Второй спринт 4](#_Toc166503250)

[2.3 Третий спринт 4](#_Toc166503251)

[2.4 Четвертый спринт 5](#_Toc166503252)

[3 Подготовка релиза 5](#_Toc166503253)

[3.1 Пятый спринт 5](#_Toc166503254)

[3.2 Шестой спринт 5](#_Toc166503255)

[3.3 Седьмой спринт 5](#_Toc166503256)

[3.4 Восьмой спринт 5](#_Toc166503257)

[4 Инструкция по использованию 5](#_Toc166503258)

[ЗАКЛЮЧЕНИЕ 6](#_Toc166503259)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 7](#_Toc166503260)

[ПРИЛОЖЕНИЕ А (рекомендуемое) Отрисовка UserFlow 8](#_Toc166503261)

[ПРИЛОЖЕНИЕ Б (рекомендуемое) Диаграмма окон 9](#_Toc166503262)

[ПРИЛОЖЕНИЕ В (рекомендуемое) Диаграмма последовательности 10](#_Toc166503263)

# ВВЕДЕНИЕ

Актуальность работы обеспечивается требованиями бизнеса. Для практически любого бизнеса нужен сайт. Такой сайт – «лицо» компании в современном мире. Без него, о бизнесе узнает гораздо меньше потенциальных клиентов, ведь они не смогут найти нужной (и что самое главное – бизнес-ориентированной) информации о нём.

Компания «Крылатые качели» – небольшая команда, специализирующаяся на проектировании и разработке детских площадок. Они уже спроектировали большое количество площадок в Абакане и собираются расширяться. Соответственно, им нужен хороший сайт. Нынешний сайт, к сожалению, не отвечает каким либо стандартам визуального дизайна и пользовательского опыта. По ссылке доступен старый сайт: <https://kacheli19.ru/.> Мне было предложено заняться разработкой нового сайта компании.

Целью создания сайта-каталога является получения новых навыков в веб-разработке и программировании, в частности, в дизайне и фронтенде. При работе с этим проектом я смогу получить опыт в full-stack разработке (разработке полного цикла) и деловой коммуникации с бизнес-заказчиком. Работа на настоящий бизнес является очень ценной для современного веб-разработчика, а опыт, полученный в работе с бизнесом, позволяет намного легче найти работу.

Для разработки были выбраны языки программирования Python[1] и Typescript. Для упрощения реализации проекта были выбраны веб-фреймворки Django[2], Django-Rest-Framework[3] и React.js[5] на Typescript[4]. Методы реализации включали модульную разработку по принципам Agile[6], разбитую на семь спринтов.

# Проектирование

1. Просмотр главной:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
2. Просмотр станицы каталога:
   1. Пользователь открывает сайт по ссылке
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на ссылку «Каталог» в хэдере страницы
   4. Сайт отображает страницу каталога с плитками тематик изделий
3. Просмотр тематики изделий:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на плитку с тематикой.
   4. Сайт отображает страницу каталога тематик c плитками категорий изделий.
4. Просмотр группы изделий:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на плитку с тематикой.
   4. Сайт отображает страницу каталога тематик c плитками категорий изделий.
   5. Пользователь нажимает на плитку категории изделия.
   6. Сайт отображает страницу каталога группы изделий с плитками изделий.
5. Просмотр изделия:
   1. Пользователь открывает сайт по ссылке.
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на плитку с тематикой.
   4. Сайт отображает страницу каталога тематик c плитками категорий изделий.
   5. Пользователь нажимает на плитку категории изделия.
   6. Сайт отображает страницу каталога группы изделий с плитками изделий.
   7. Пользователь нажимает на плитку изделия.
   8. Сайт отображает страницу информации об изделии.
6. Просмотр страницы контактов:
   1. Пользователь открывает сайт по ссылке
   2. Сайт отображает главную с каталогом тематик изделий и качествами компании
   3. Пользователь нажимает на ссылку «Контакты» в хэдере страницы
   4. Сайт отображает страницу контактов

Далее показана диаграмма использования (рисунок 1)

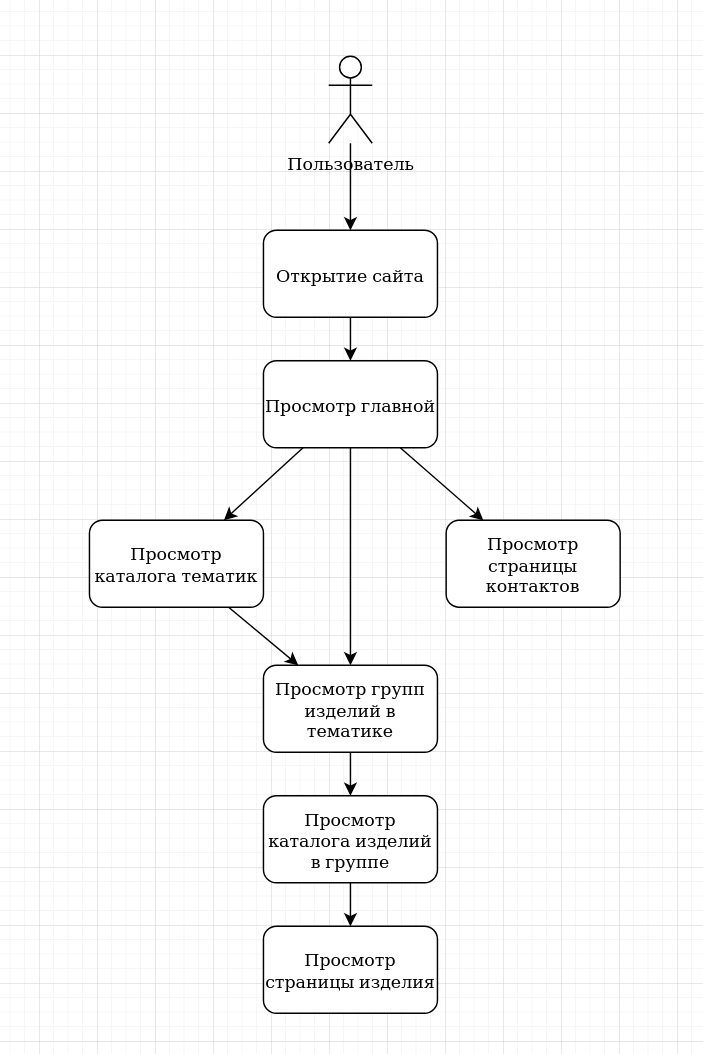


Рисунок 1 – Диаграмма вариантов использования

Диаграмма деятельности (рисунок 2).

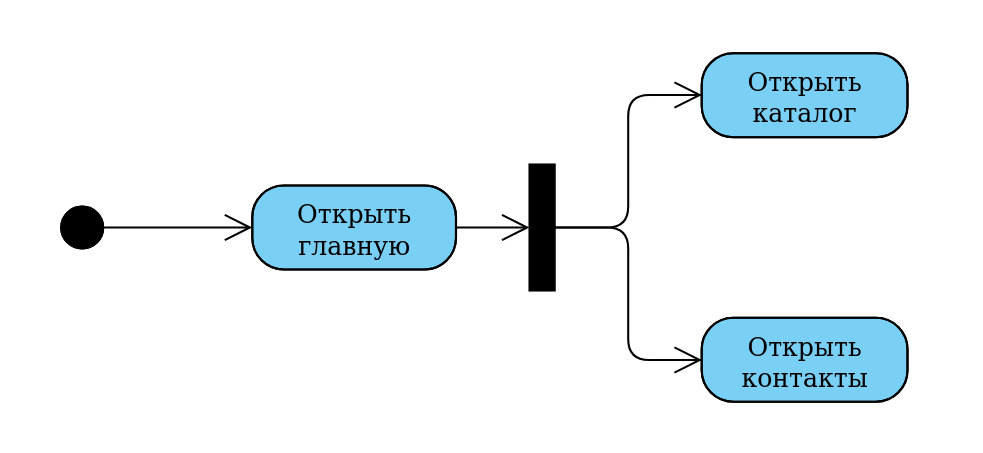


Рисунок 2 – Диаграмма вариантов использования

Шаблоны проектирования, использующиеся в приложении:

* MVC(Model-View-Controller). Используется по умолчанию в Django;
* Singleton. Используется в Django ORM для подключения к базе данных;
* Decorator. Используется в Django-Rest-Framework.
* FSD (Feature-Sliced Design). Архитектурная методология для frontend-приложений

# Разработка минимального стоящего продукта

Минимально стоящий продукт (MVP) проекта представляет из себя веб-сайт визитку с каталогом. В каталоге есть три сущности:

* тематика изделий;
* группа изделий;
* изделие.

Тематика изделий содержит в себе группы изделий, группы – сами изделия. Эти сущности хранятся в базе данных и конфигурируются из админ панели Django.

## Первый спринт

Список задач на спринт приведён в таблице 1.

Таблица 1 – задачи спринта 1

|  |  |  |
| --- | --- | --- |
| **№** | **Оценка** | **Задача** |
| 1 | 1 | Найти референсы с сайтов конкурентов. |
| 2 | 2 | Создать черновик дизайна в виде общих форм и расположений элементов. |
| 3 | 6 | Составить сущности в базе данныхать дизайн в чёрно-белой расцветке. |
| 4 | 1 | Создать палитру цветов на основе рекламных карточек компании. |
| 5 | 3 | Разработать дизайн в цвете палитры. |
| 6 | 1 | Согласовать с заказчиком. |
| Sum | 14 |  |

### **Результаты выполнения задач**

Результаты выполнения задач представлены на рисунках с 3 по 8.

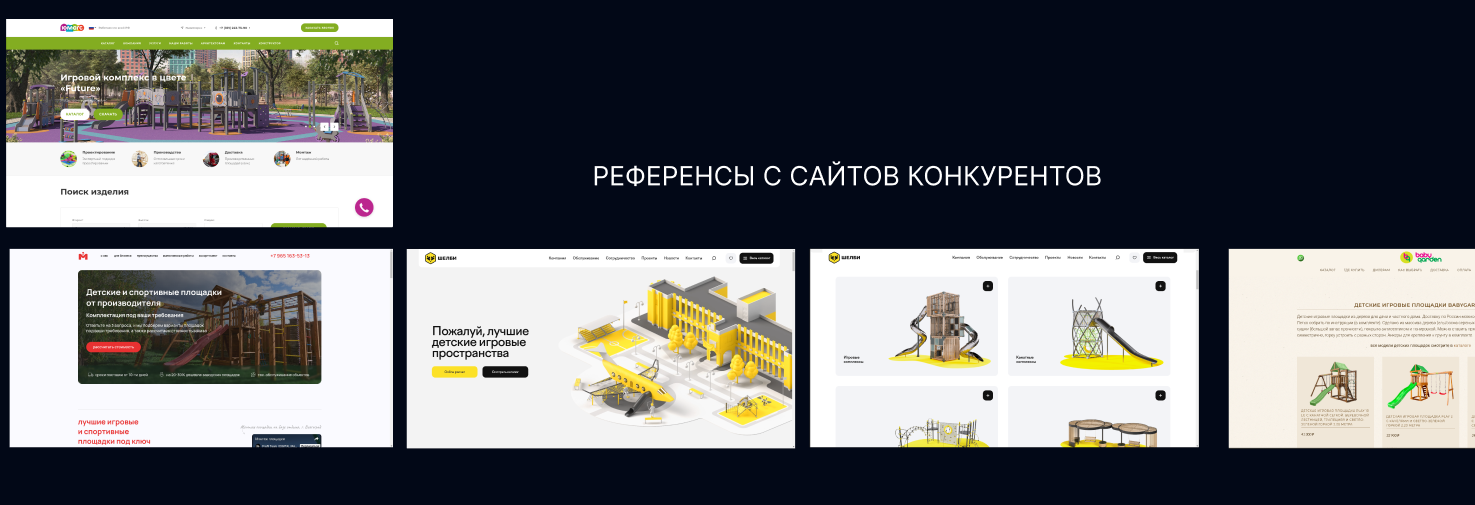


Рисунок 1 – Референсы с сайта конкурентов

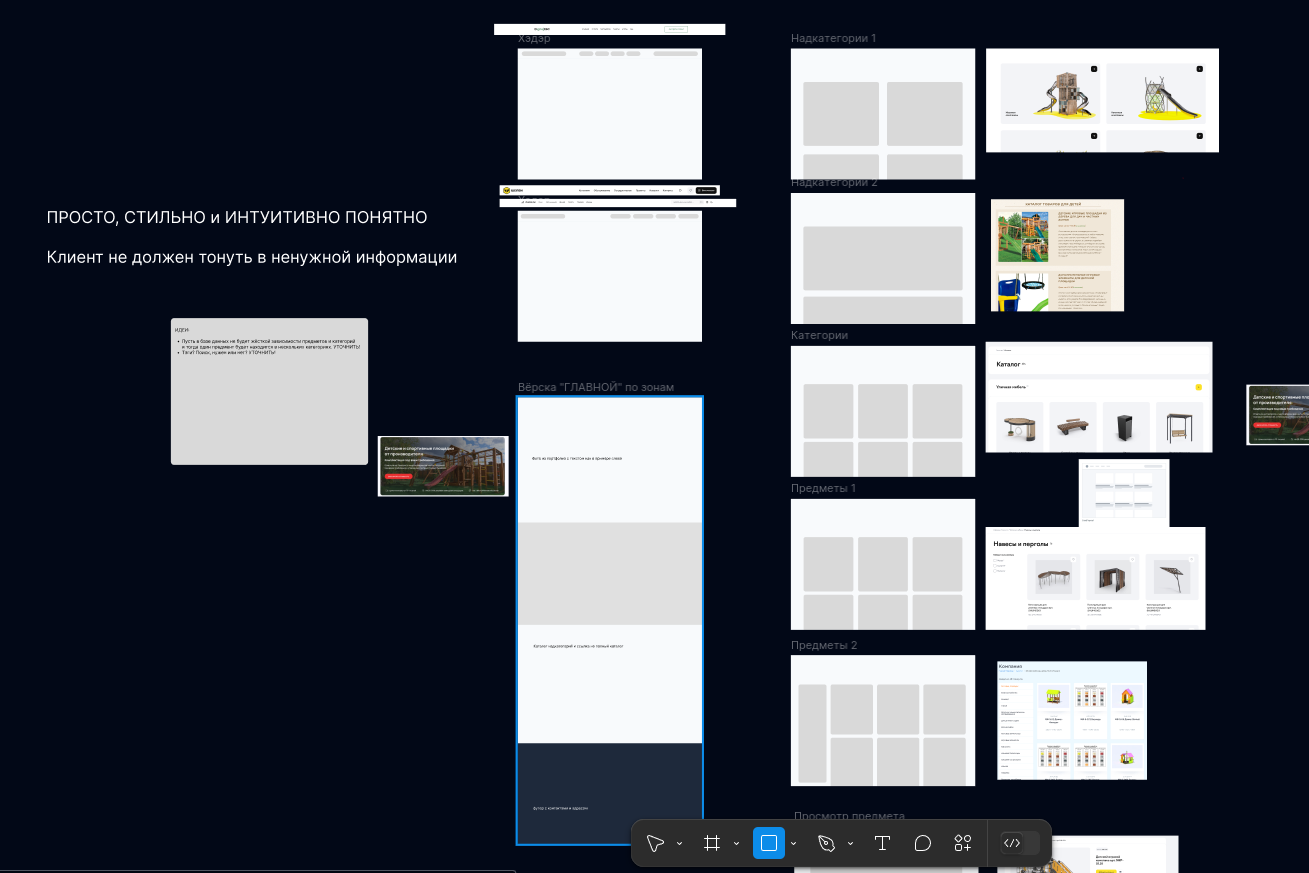


Рисунок 2 – Набросок дизайна составленный на основе референсов

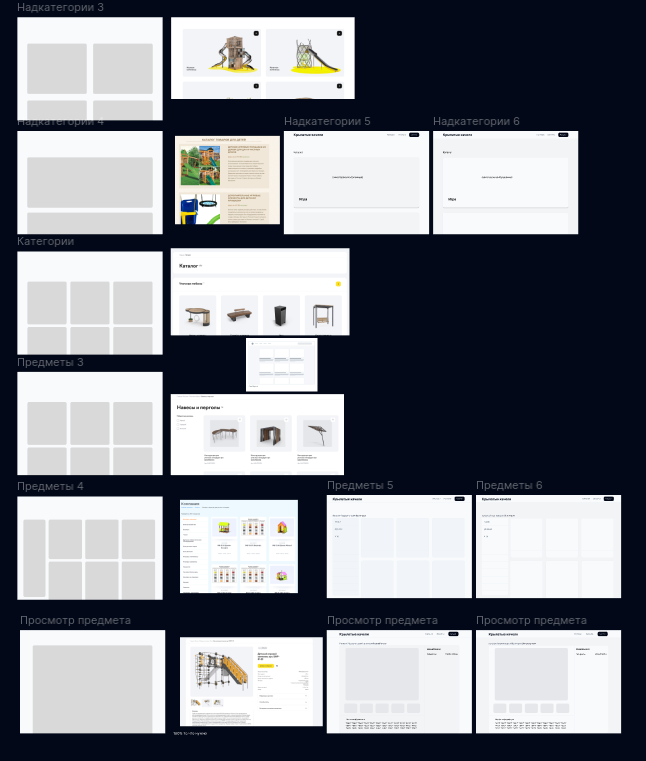


Рисунок 3 – Черновик дизайна каталога



Рисунок 4 – Палитра цветов

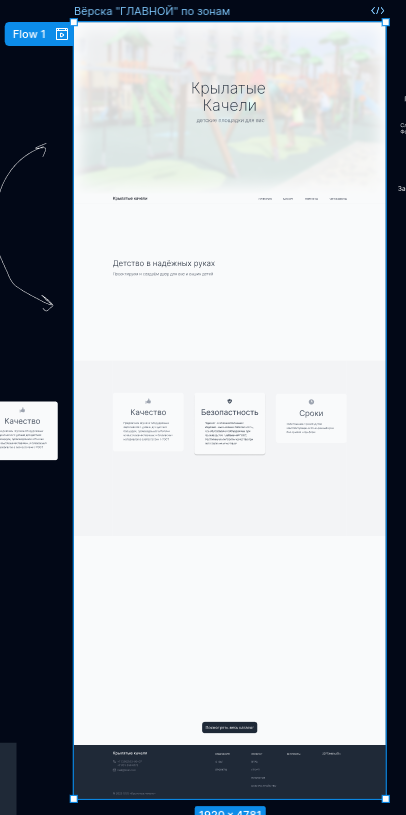


Рисунок 5 – Дизайн главной страницы в работе



Рисунок 6 – Готовая административная панель

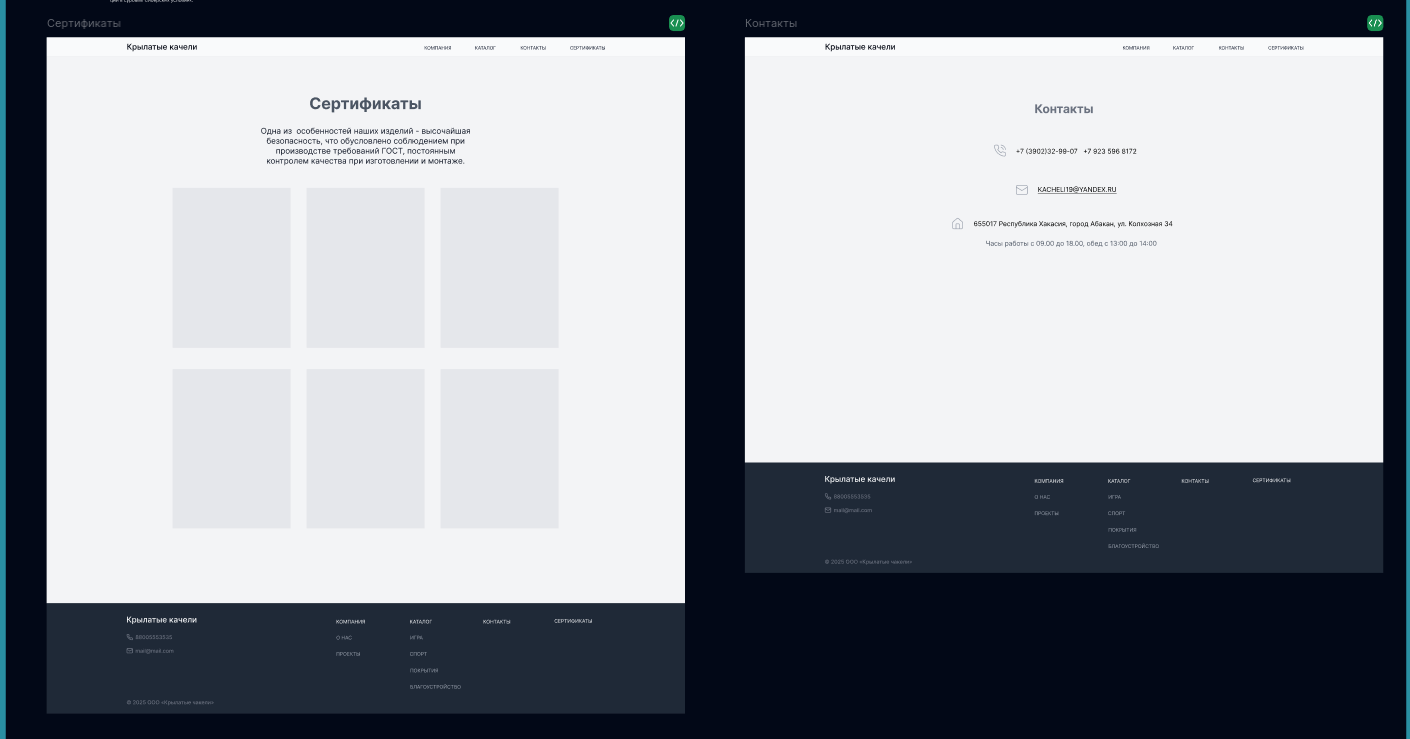


Рисунок 7 – Готовые дизайны страниц сертификатов и контактов

Рисунок 8 – Диаграмма сгорания задач

## Второй спринт

Список задач на спринт приведён в таблице 2.

Таблица 2 – задачи спринта 2.

|  |  |  |
| --- | --- | --- |
| **№** | **Оценка** | **Задача** |
| 1 | 1 | Создать Djnago + vite проект. |
| 2 | 1 | Составить сущности в базе данных |
| 3 | 2 | Организовать CI/CD |
| 4 | 4 | Организовать работу pre-commits |
| 5 | 1 | Базовый CRUD для моделей |
| Sum | 9 |  |

### Итоги текущего спринта

Результаты выполнения задач представлены на рисунках с 9 по 19

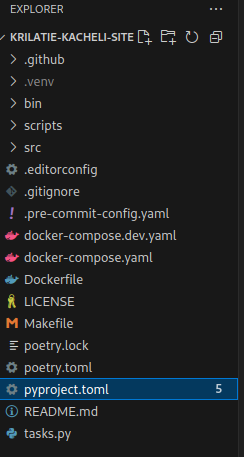


Рисунок 9 – Структура проекта

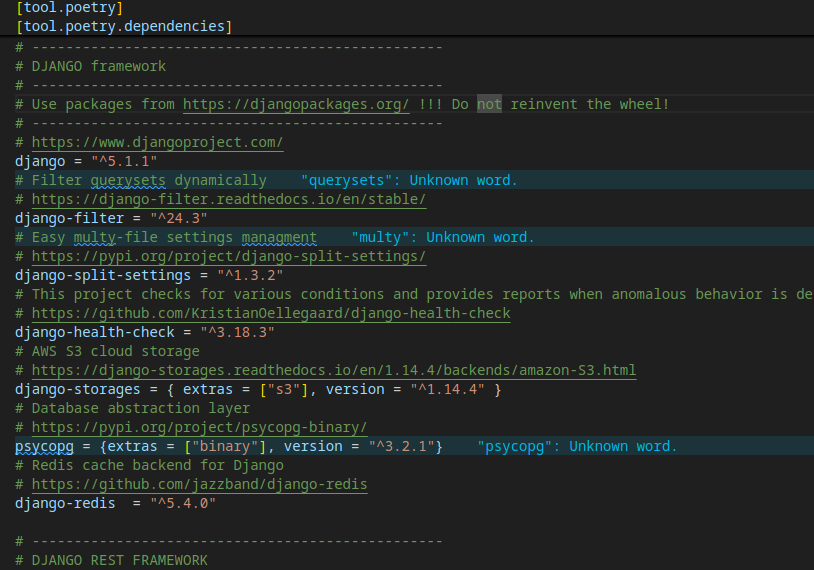


Рисунок 10 – Файл pyproject.toml

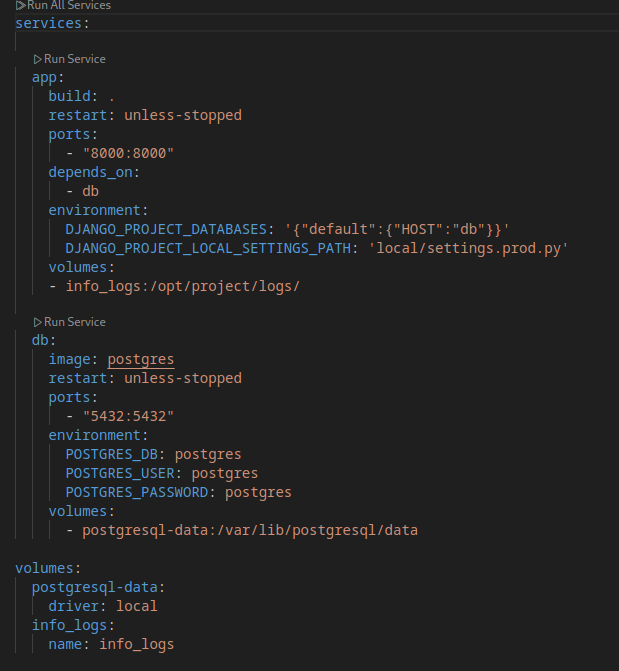


Рисунок 11 – Файл docker-compose.yaml

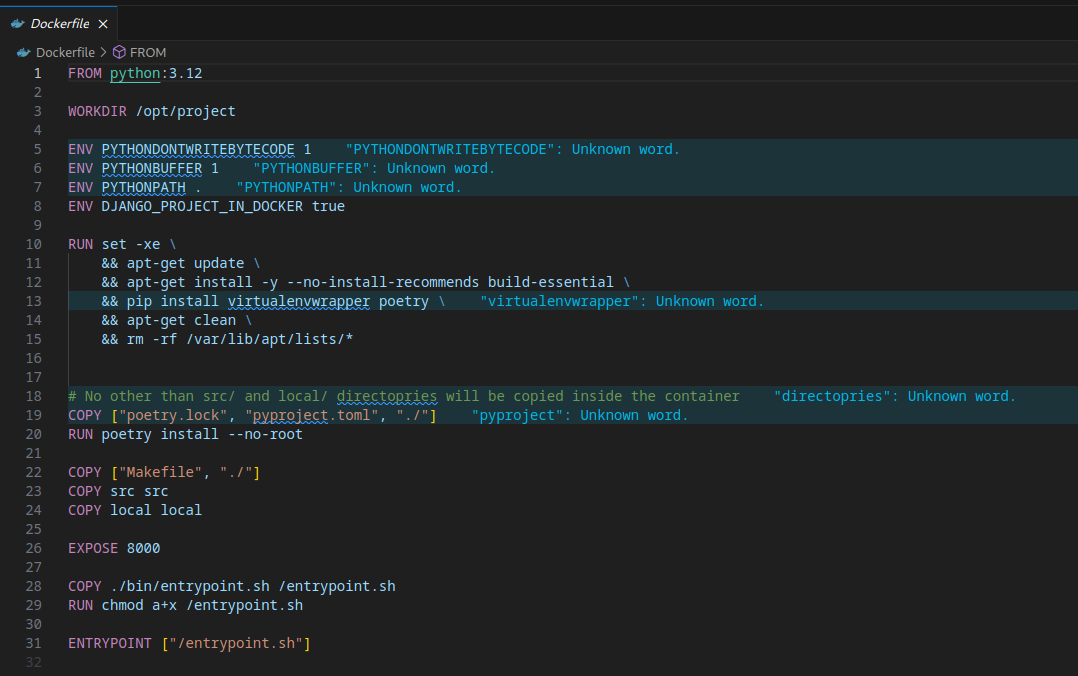


Рисунок 12 – Файл Dockerfile

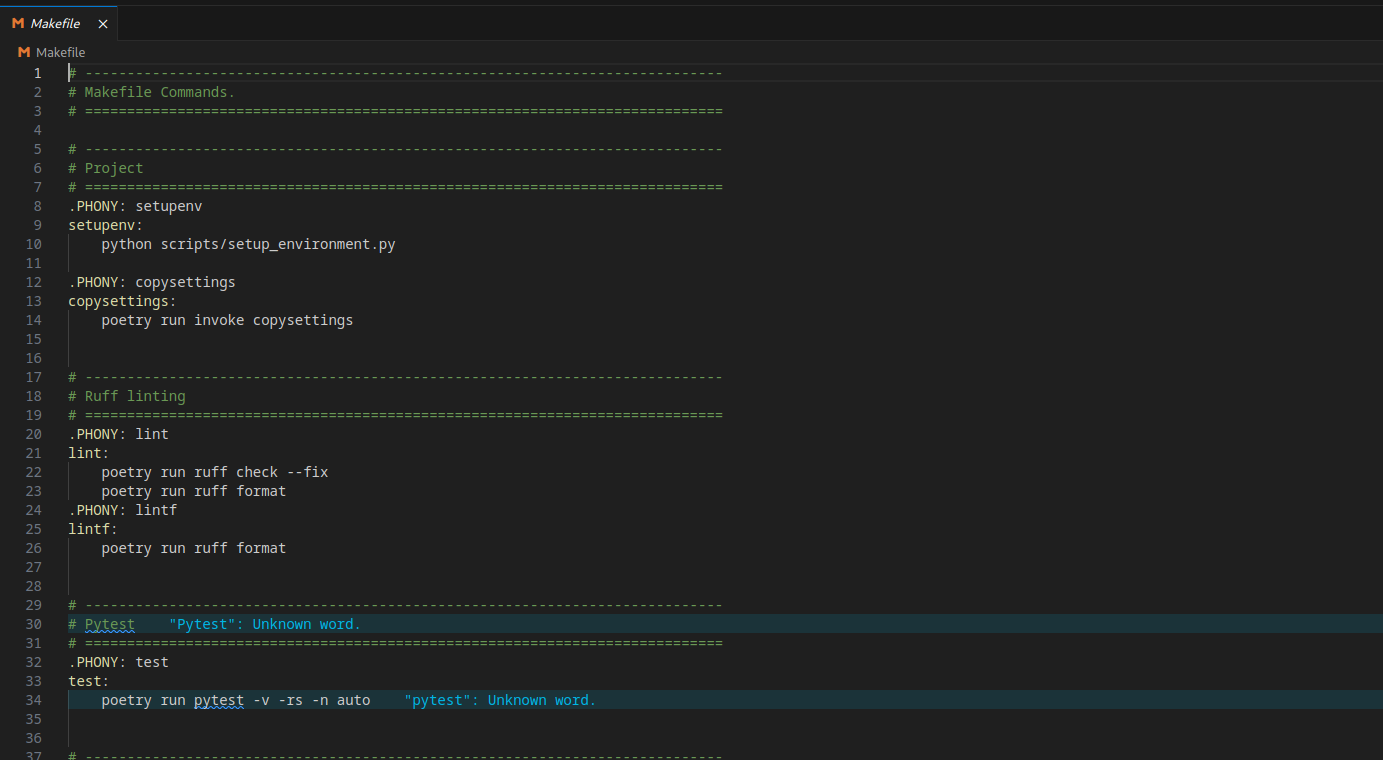


Рисунок 13 – Файл Makefile

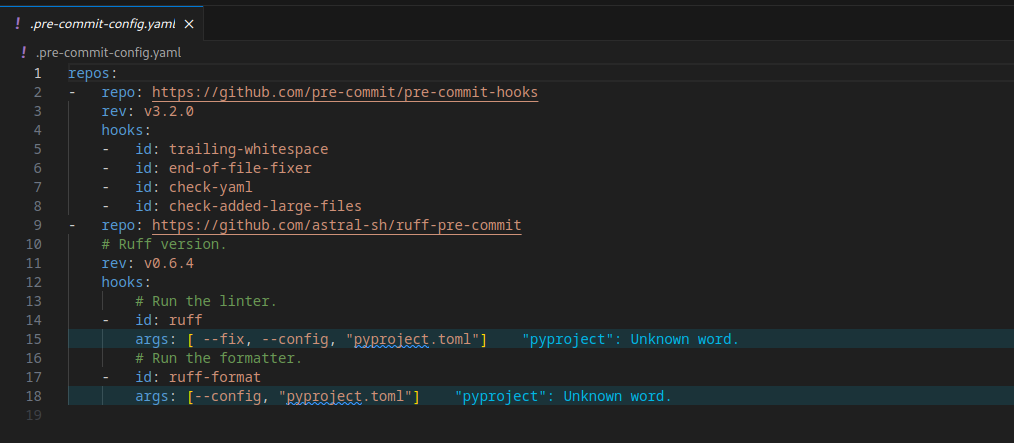


Рисунок 14 – Файл pre-commits

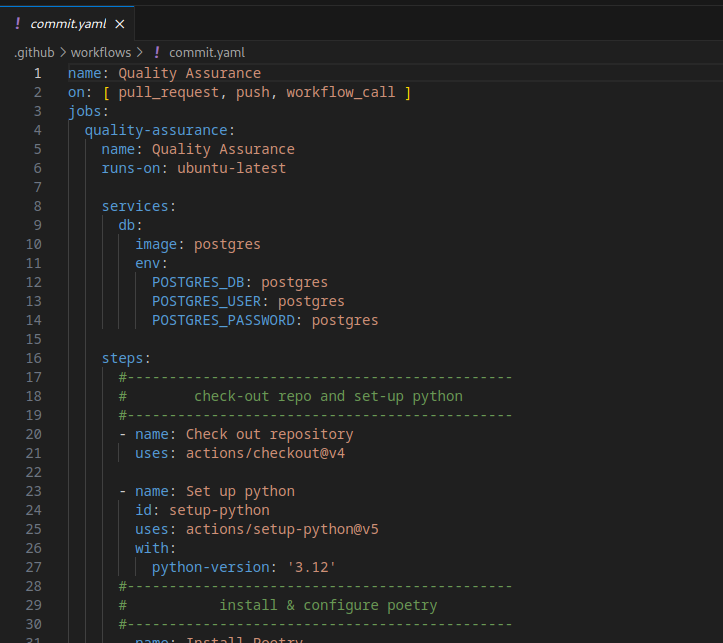


Рисунок 15 – Настройка github actions

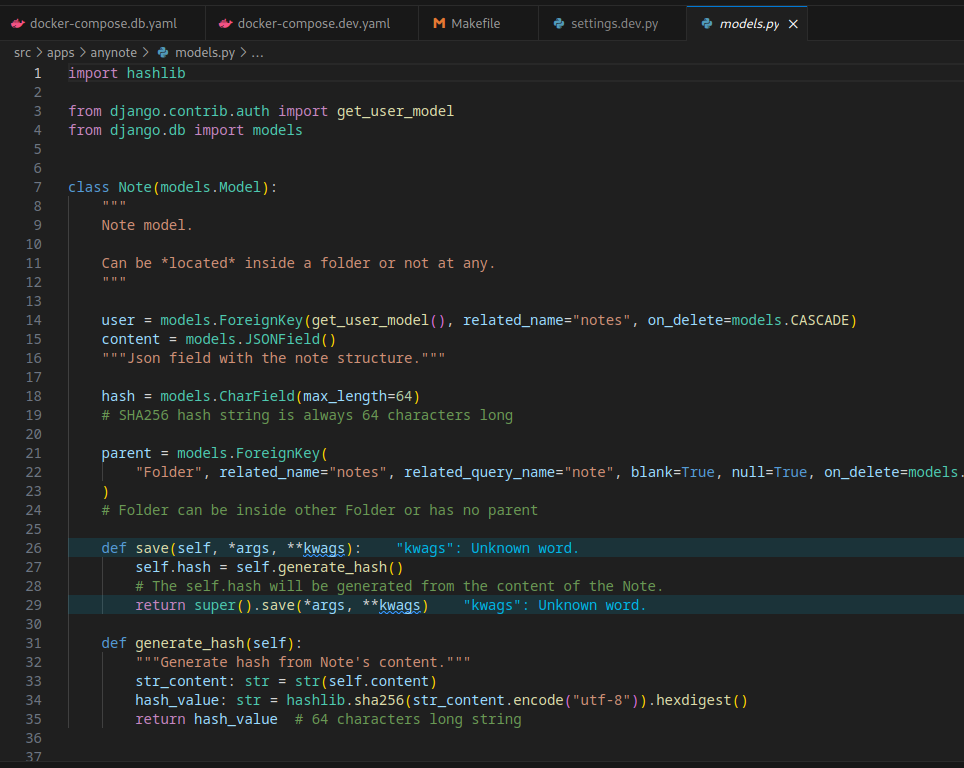


Рисунок 16 – Модели Django

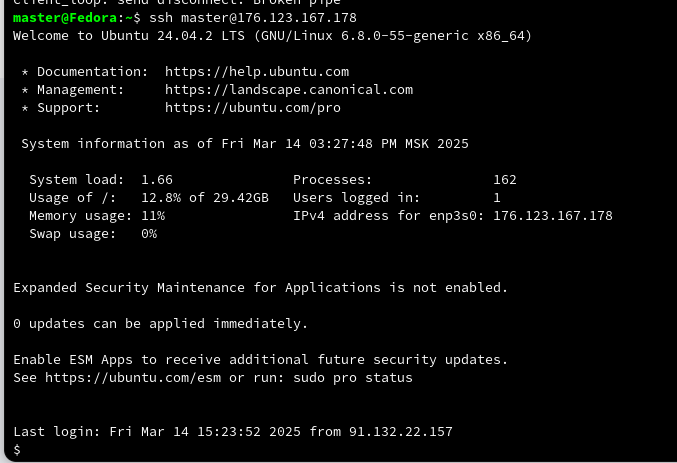


Рисунок 17 – Виртуальная машина доступна для доступа

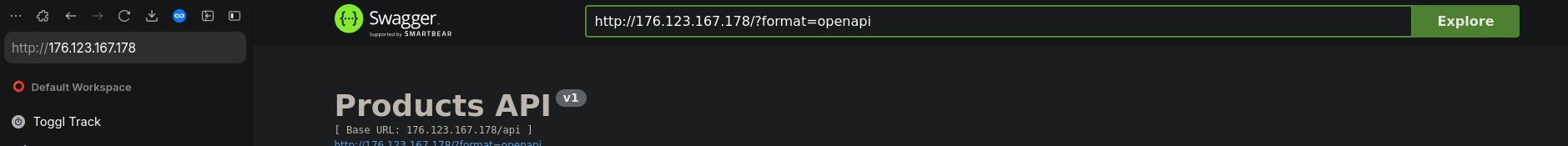


Рисунок 18 – Сайт на виртуальной машине

Рисунок 19 – Диаграмма сгорания задач

## Третий спринт

Список задач на спринт приведён в таблице 3.

Таблица 3 – Планы на спринт 3

|  |  |  |
| --- | --- | --- |
| **№** | **Оценка** | **Задача** |
| 1 | 1 | Согласовать правки в UI/UX сайта |
| 2 | 2 | Внести правки в главную |
| 3 | 1 | Преобразовать визуальные ресурсы заказчика к дизайну сайта |
| 4 | 2 | Внести правки в каталог |
| 5 | 1 | Доработать макет в используя цветовую палитру |
| Sum | 7 |  |

### Итоги текущего спринта

Результаты выполнения задач представлены на рисунках с 20 по 29.

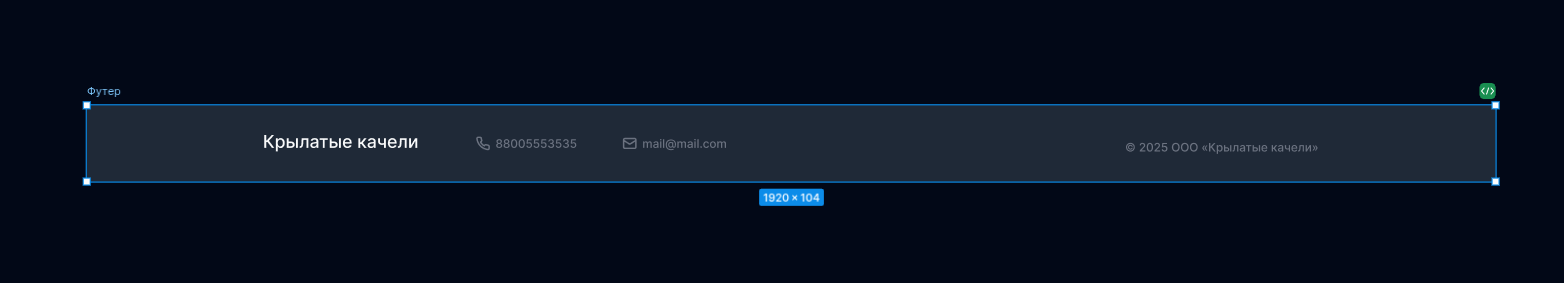


Рисунок 20 – Уменьшенный футер



Рисунок 21 – Фото одной из страниц каталога

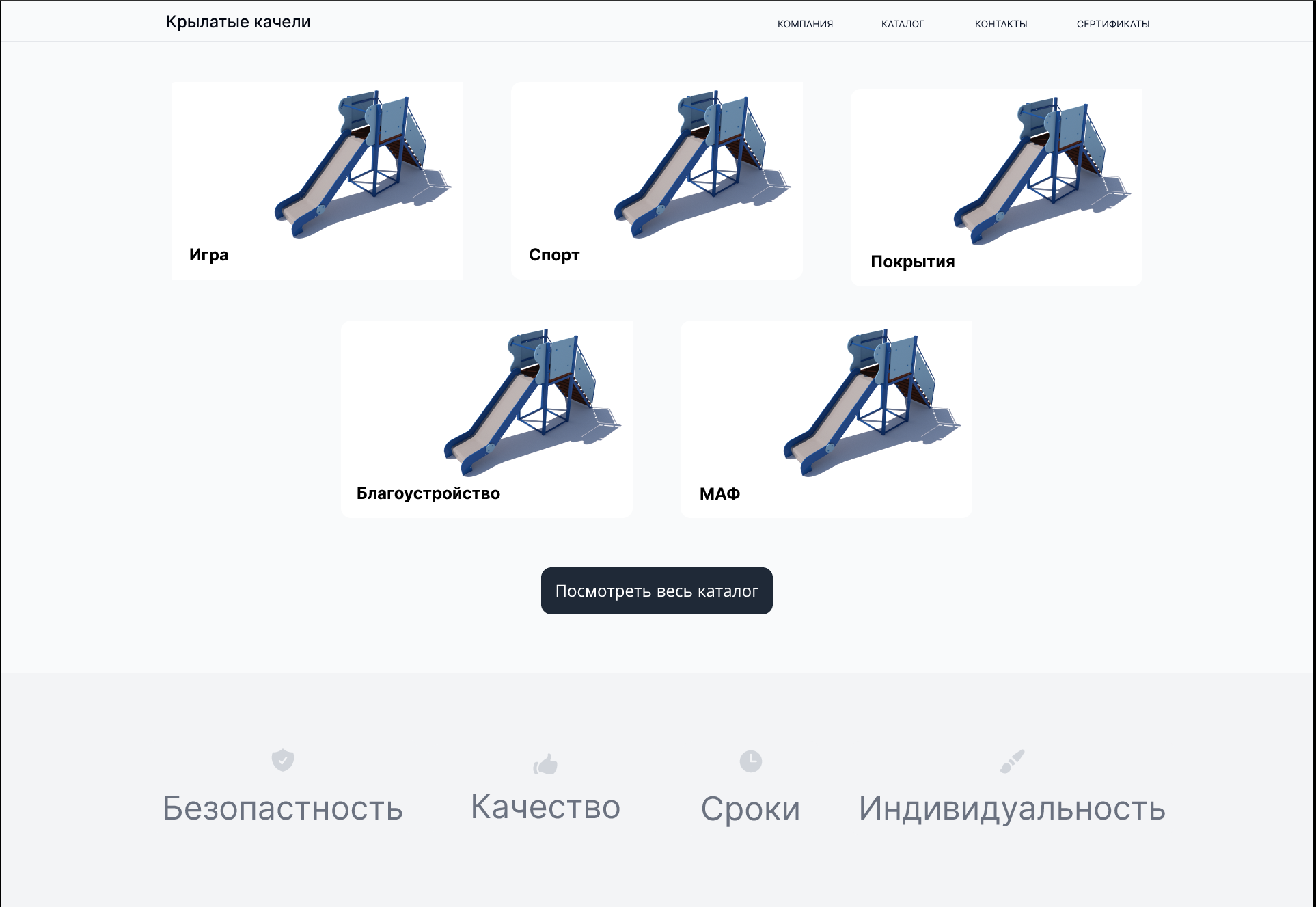


Рисунок 22 – Изменённая главная страница

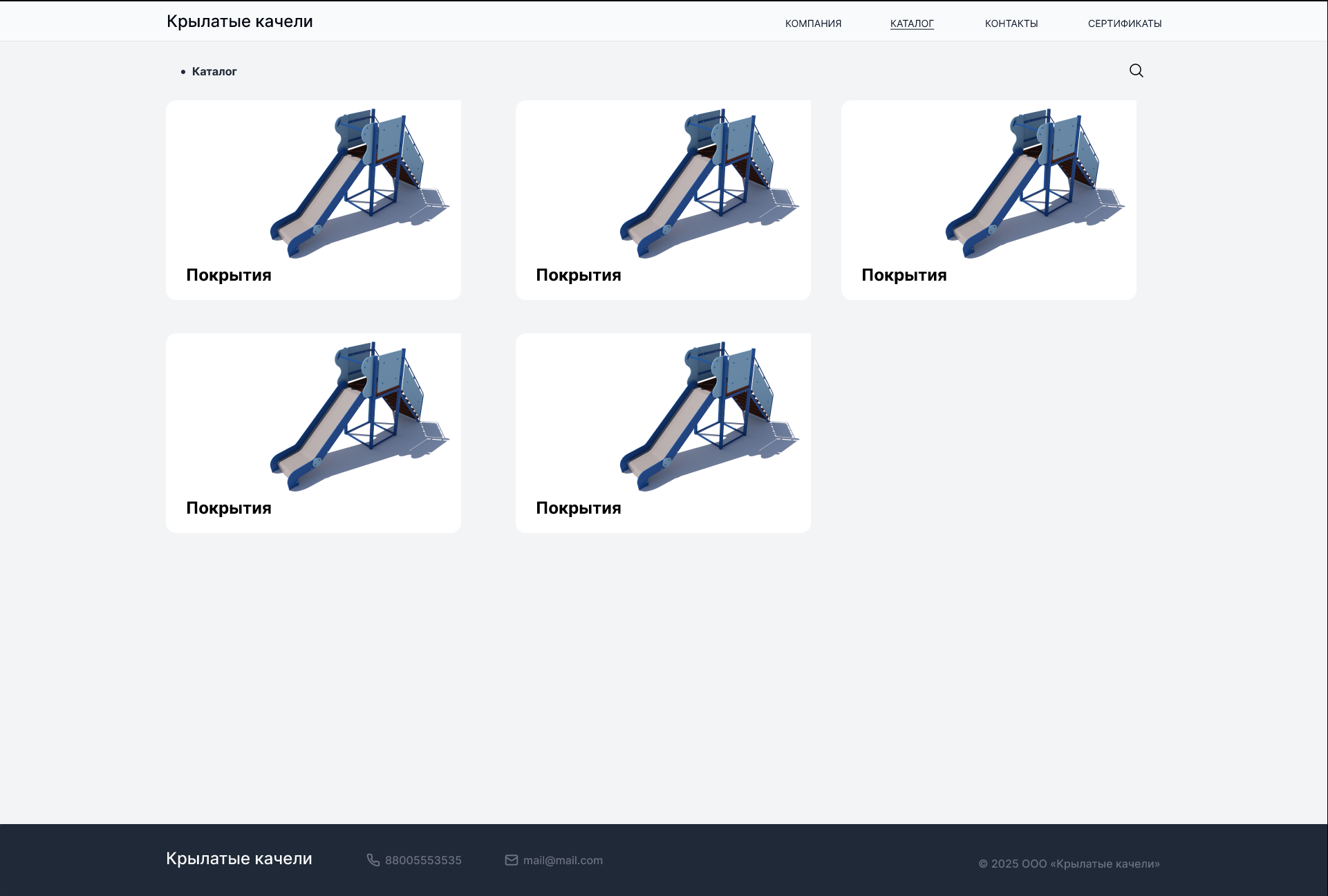


Рисунок 23 – Каталог надкатегорий изделий

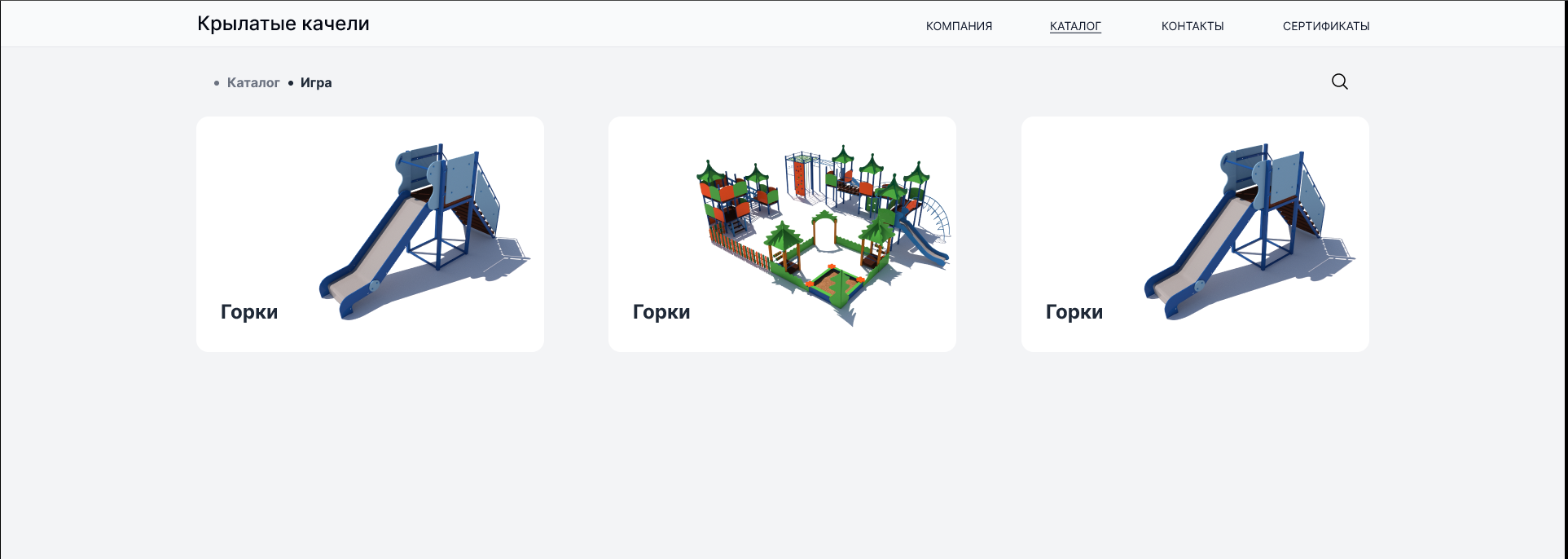


Рисунок 24 – Каталог категорий

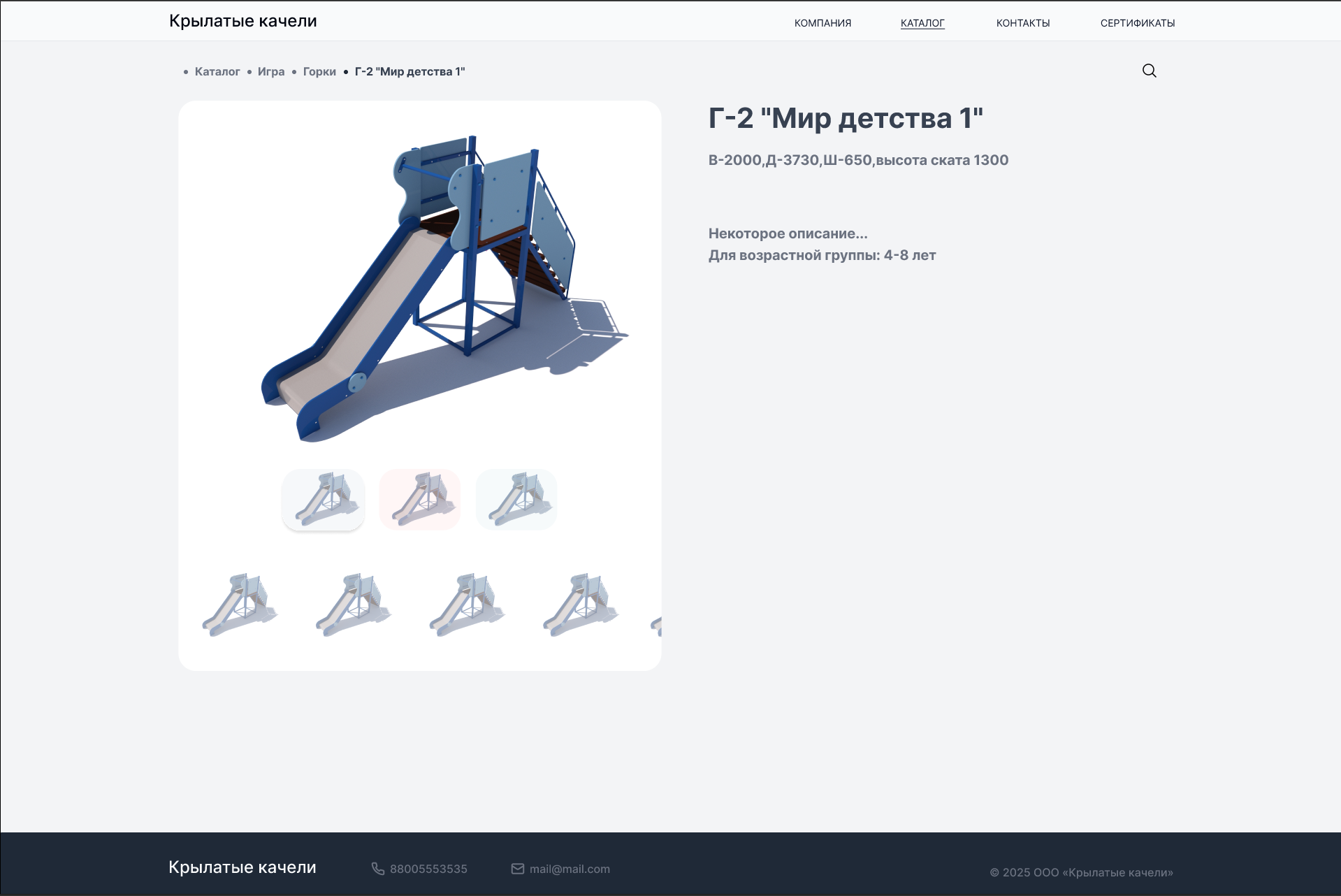


Рисунок 25 – Страница изделия

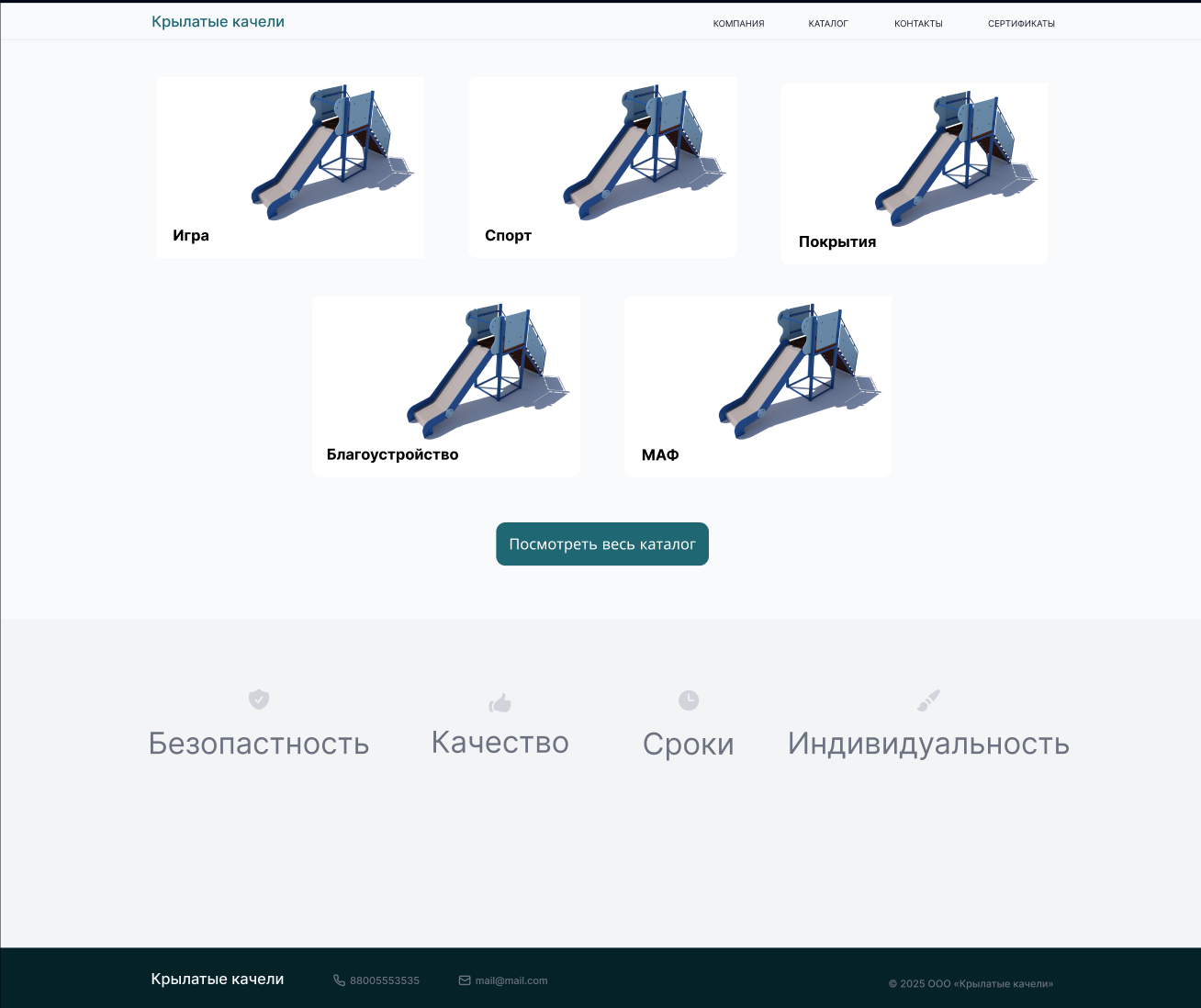


Рисунок 26 – Главная в цвете

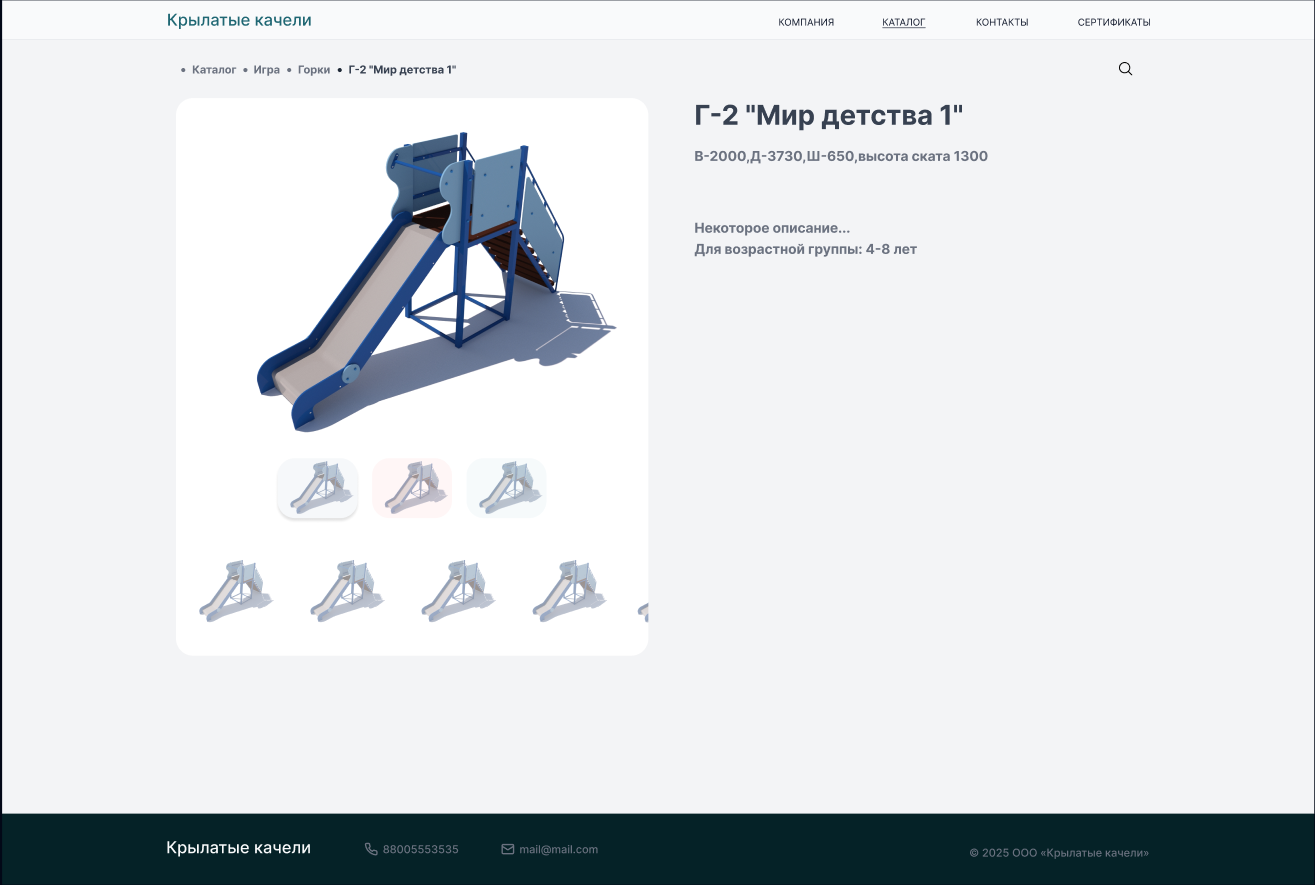


Рисунок 27 – Каталог в цвете

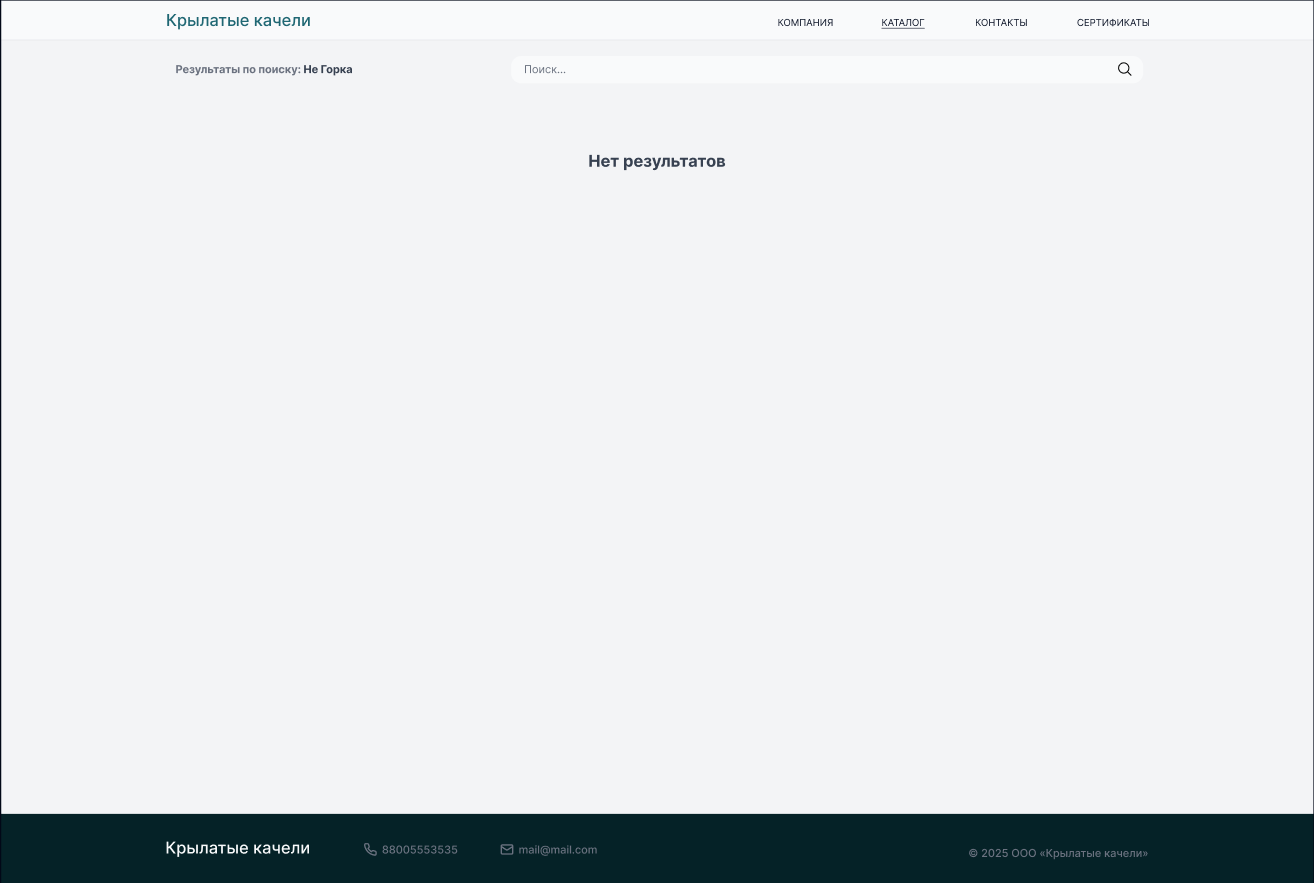


Рисунок 28 – Новая страница поиска

Рисунок 29 – Диаграмма сгорания задач

## Четвертый спринт

Список задач на спринт приведён в таблице 4.

Таблица 4 – Планы на спринт 4

|  |  |  |
| --- | --- | --- |
| **№** | **Оценка** | **Задача** |
| 1 | 1 | Добавить приложение каталога |
| 2 | 1 | Сделать модели Изделия, Группы и Тематики |
| 3 | 1 | Сделать сериализаторы для моделей |
| 4 | 2 | Настроить роутинг каталога и подключить приложение |
| 5 | 1 | Был написан CRUD для моделей |
| 6 | 1 | Подключить дб и сделать миграции |
| Sum | 7 |  |

### Итоги текущего спринта

Результаты выполнения задач представлены на рисунках с 30 по 37.

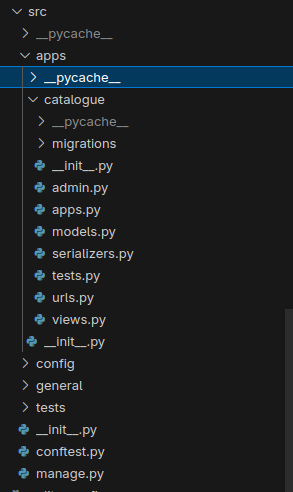


Рисунок 30 – Структура приложение каталога

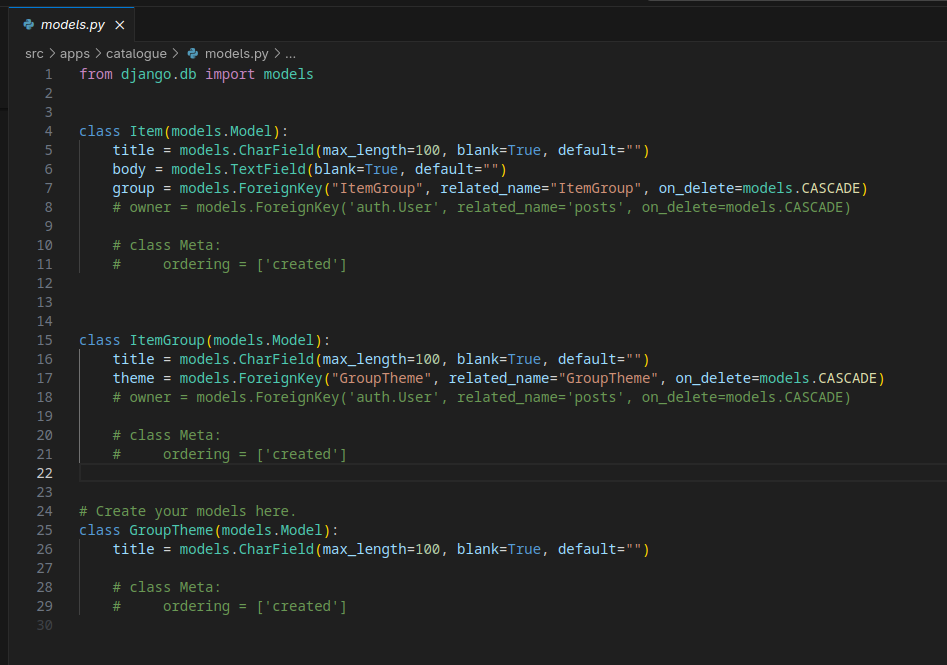


Рисунок 31 – Модели приложения

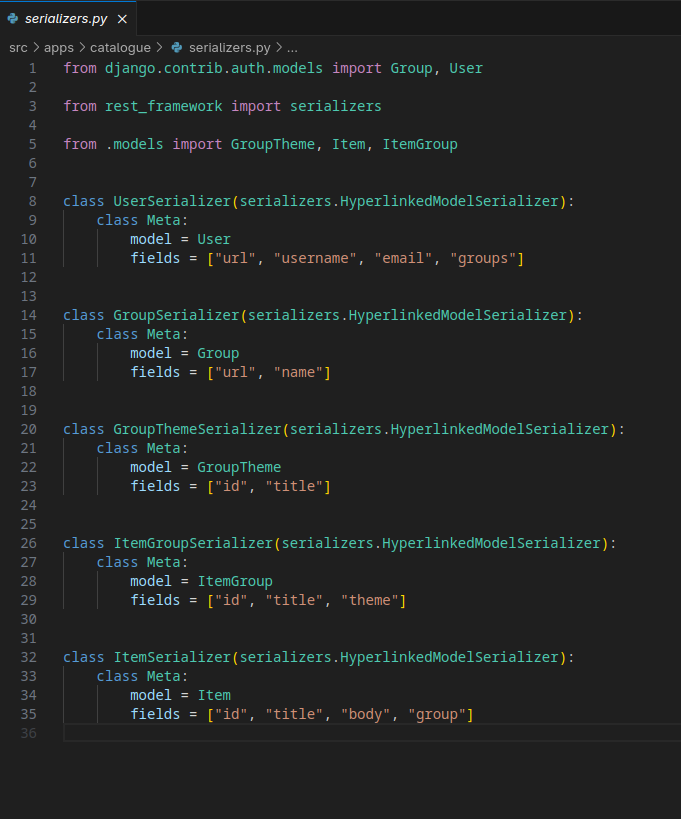


Рисунок 32 – Сериализаторы для моделей

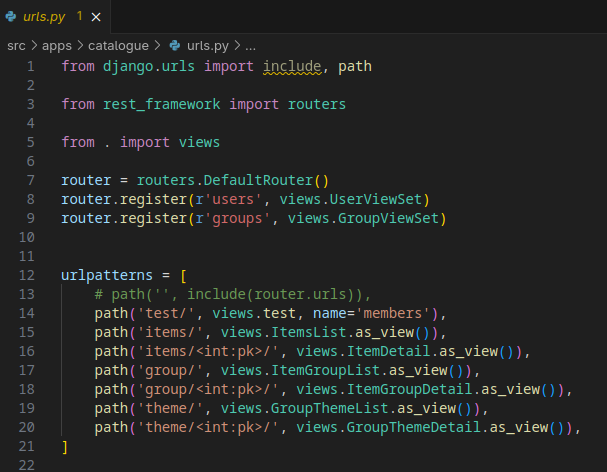


Рисунок 33 – Каталог надкатегорий изделий

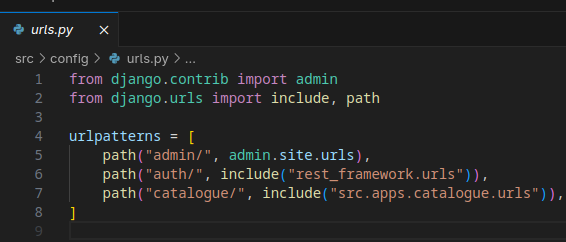


Рисунок 34 – Каталог категорий

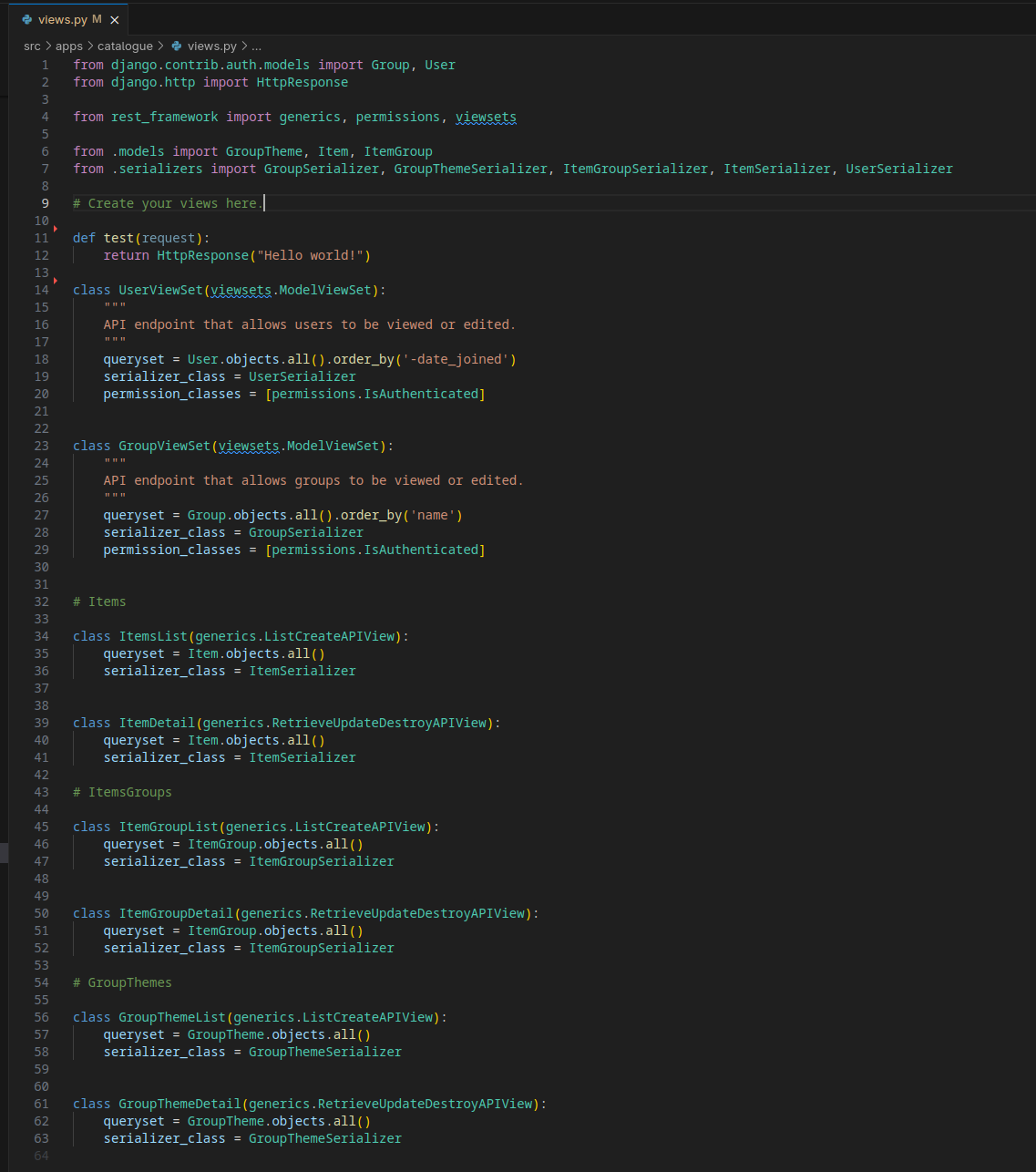


Рисунок 35 – содержимое views.py

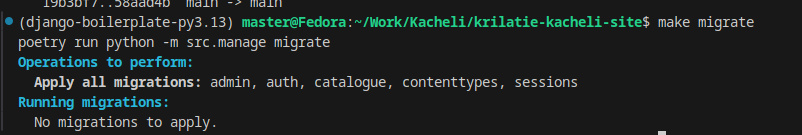


Рисунок 36 – Миграции уже проведены

Рисунок 37 – Диаграмма сгорания задач

# Подготовка релиза

За предыдущие четыре спринта были реализованы бэкенд и дизайн. Далее следует фронтенд, связывание бэкенда и фронтенда и загрузка на сервер.

## Пятый спринт

Список задач на спринт приведён в таблице 5.

Таблица 5 – Планы на спринт 5

|  |  |  |
| --- | --- | --- |
| **№** | **Оценка** | **Задача** |
| 1 | 3 | Изучить React |
| 2 | 1 | Создать React + vite приложение |
| 3 | 1 | Применить FSD архитектуру |
| 4 | 2 | Сверстать главную страницу |
| Sum | 7 |  |

### Итоги текущего спринта

Результаты выполнения задач представлены на рисунках с 38 по 45

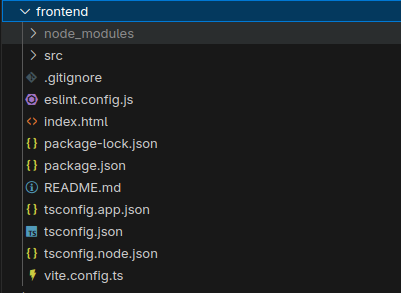


Рисунок 38 – Структура приложение каталога

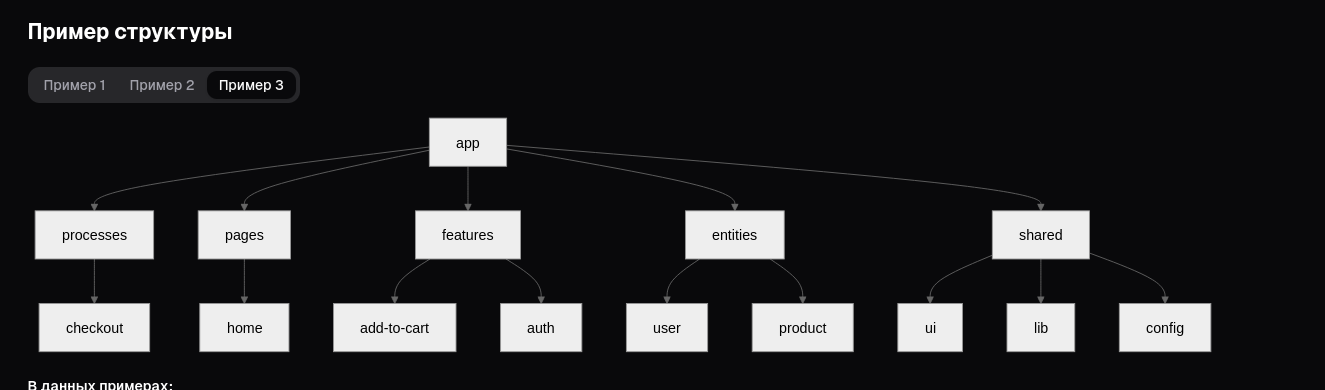


Рисунок 39 – Пример реализации FSD архитектуры

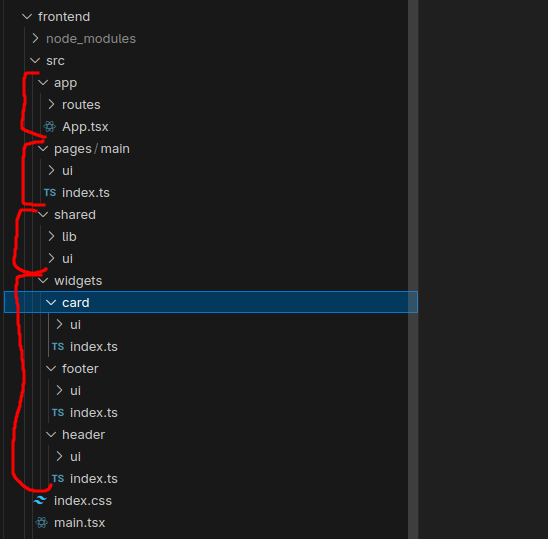


Рисунок 40 – Логические слои приложения

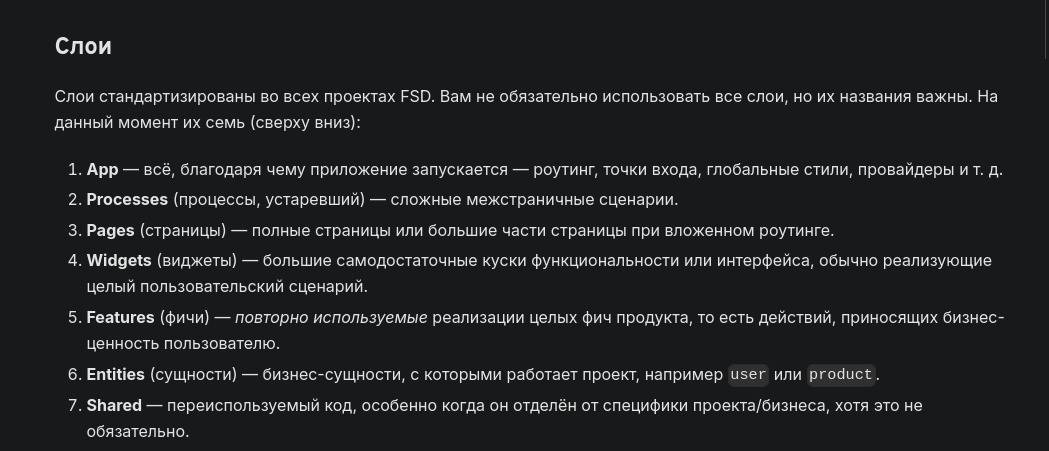


Рисунок 41 – Фото документации

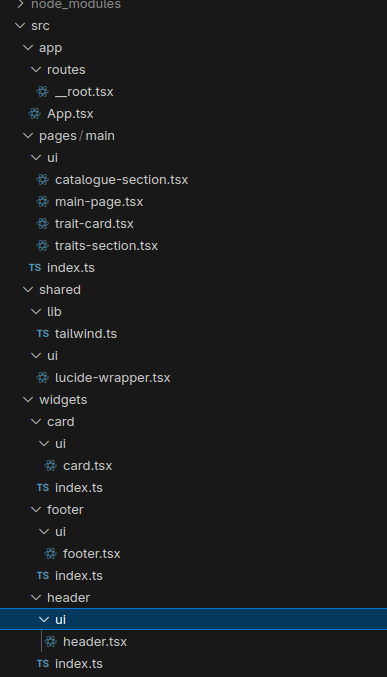


Рисунок 42 – Полная структура проекта.

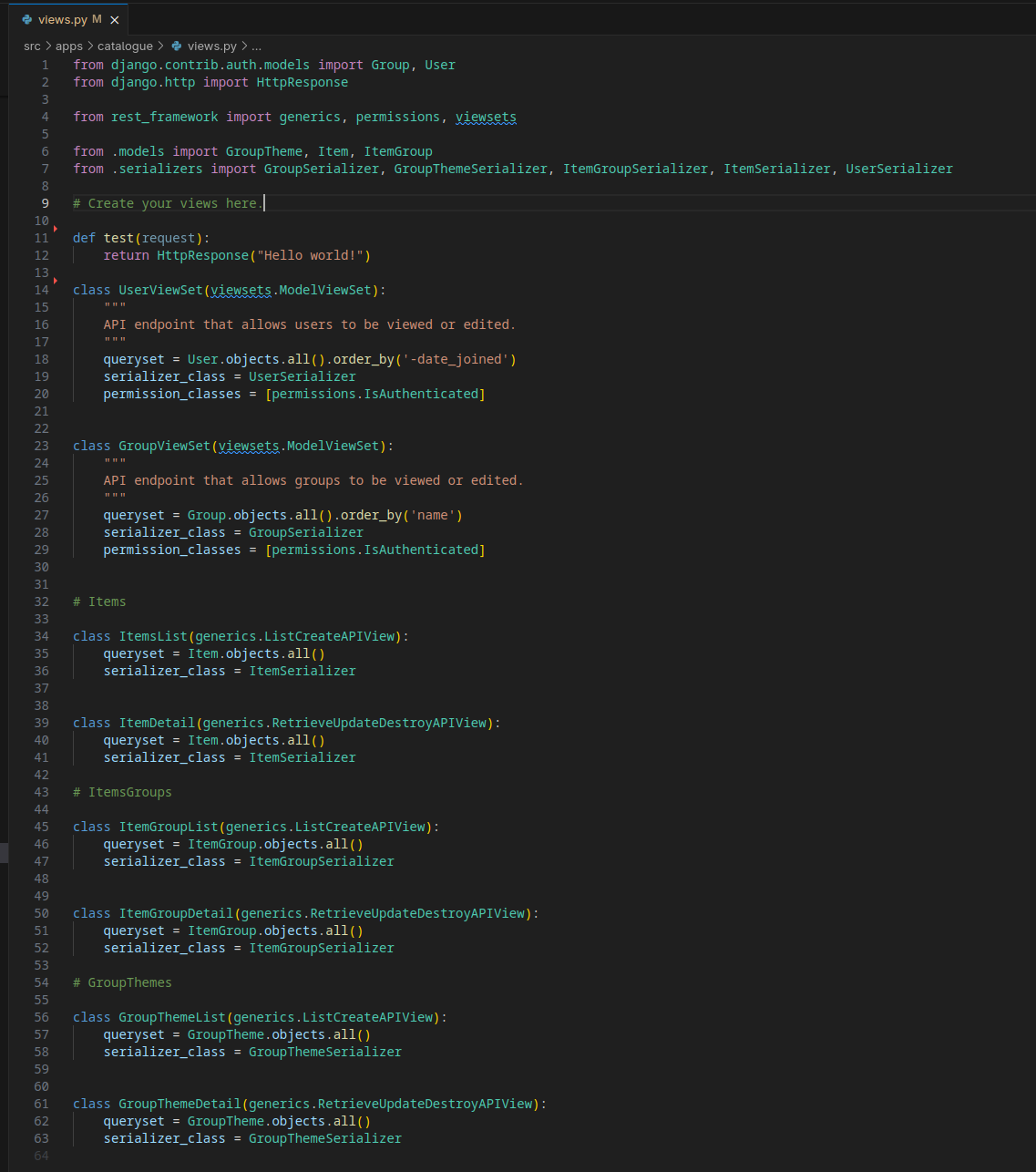


Рисунок 43 – содержимое views.py

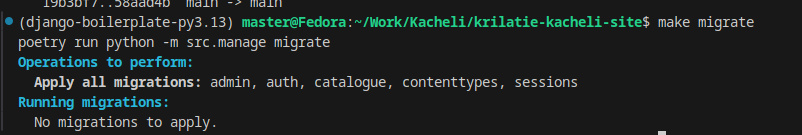


Рисунок 44 – Миграции уже проведены

Рисунок 45 – Диаграмма сгорания задач

## Шестой спринт

Список задач на спринт приведён в таблице 6.

Таблица 6 – Планы на спринт 6

|  |  |  |
| --- | --- | --- |
| **№** | **Оценка** | **Задача** |
| 1 | 3 | Связать фронтенд с бэкендом отправкой запросов |
| 2 | 2 | Доработать систему отображения элементов каталога для работы с бэкендом |
| 3 | 1 | Настроить CORS headers |
| 4 | 1 | Стилистические правки в фронтенде |
| Sum | 7 |  |

### Итоги текущего спринта

Результаты выполнения задач представлены на рисунках с 46 по 52



Рисунок 46 – API методы для работы с бэкендом

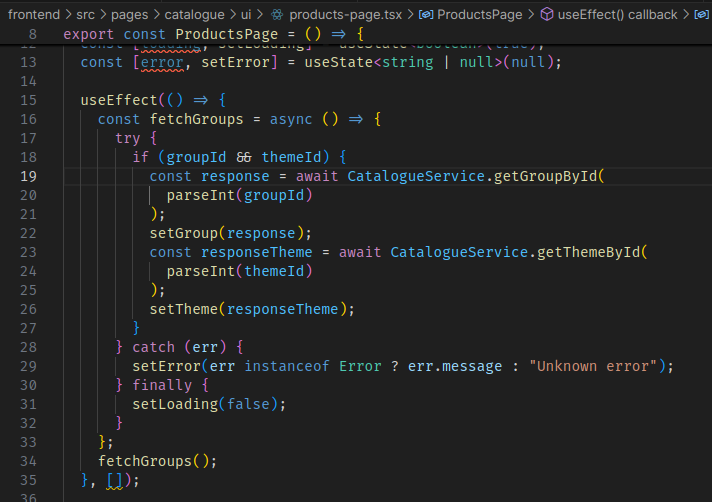


Рисунок 47 – Вызов методов API при загрузке страницы

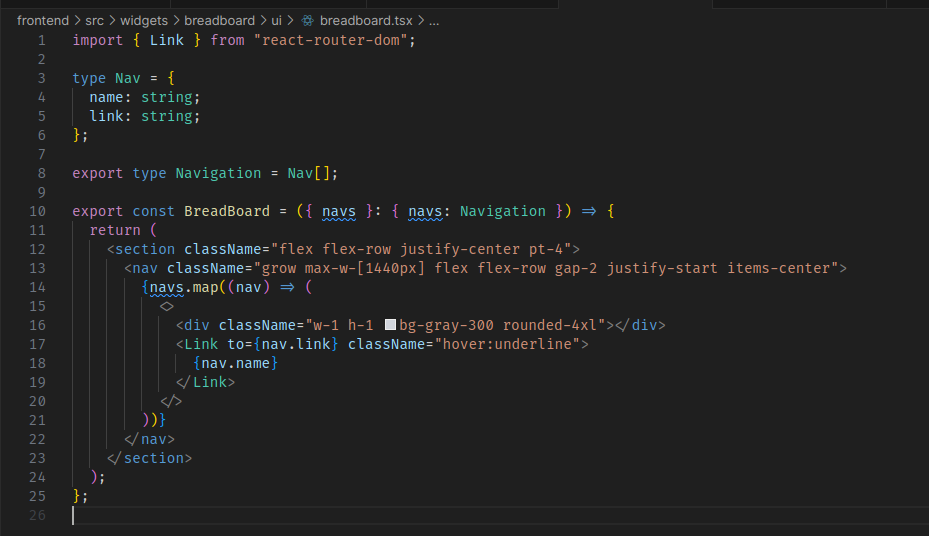


Рисунок 48 – Элемент отображающий путь до элемента каталога

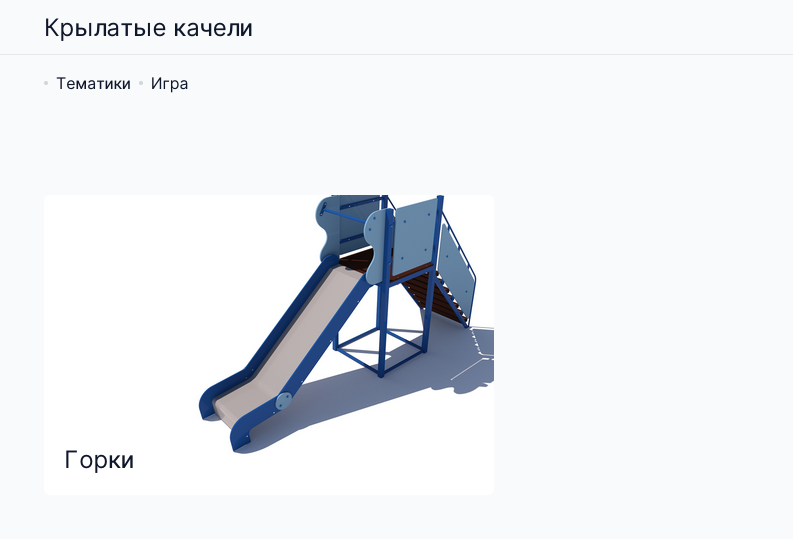


Рисунок 49 – Элемент отображающий путь наглядно

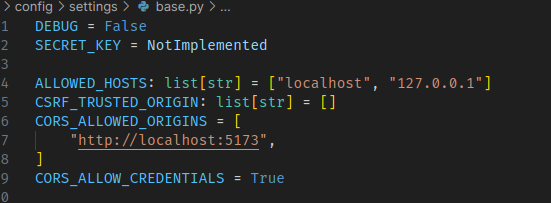


Рисунок 50 – Временная настройка CORS для разработки.

Рисунок 51 – Диаграмма сгорания задач

## Седьмой спринт

Список задач на спринт приведён в таблице 7.

Таблица 7 – Планы на текущий спринт

|  |  |  |
| --- | --- | --- |
| **№** | **Оценка** | **Задача** |
| 1 | 1 | Исправления фронтенда |
| 3 | 2 | Настройка виртуальной машины |
| 4 | 4 | Развёртка на виртуальной машине |
| Sum | 7 |  |

### Итоги текущего спринта

Результаты выполнения задач представлены на рисунках с 52 по 58

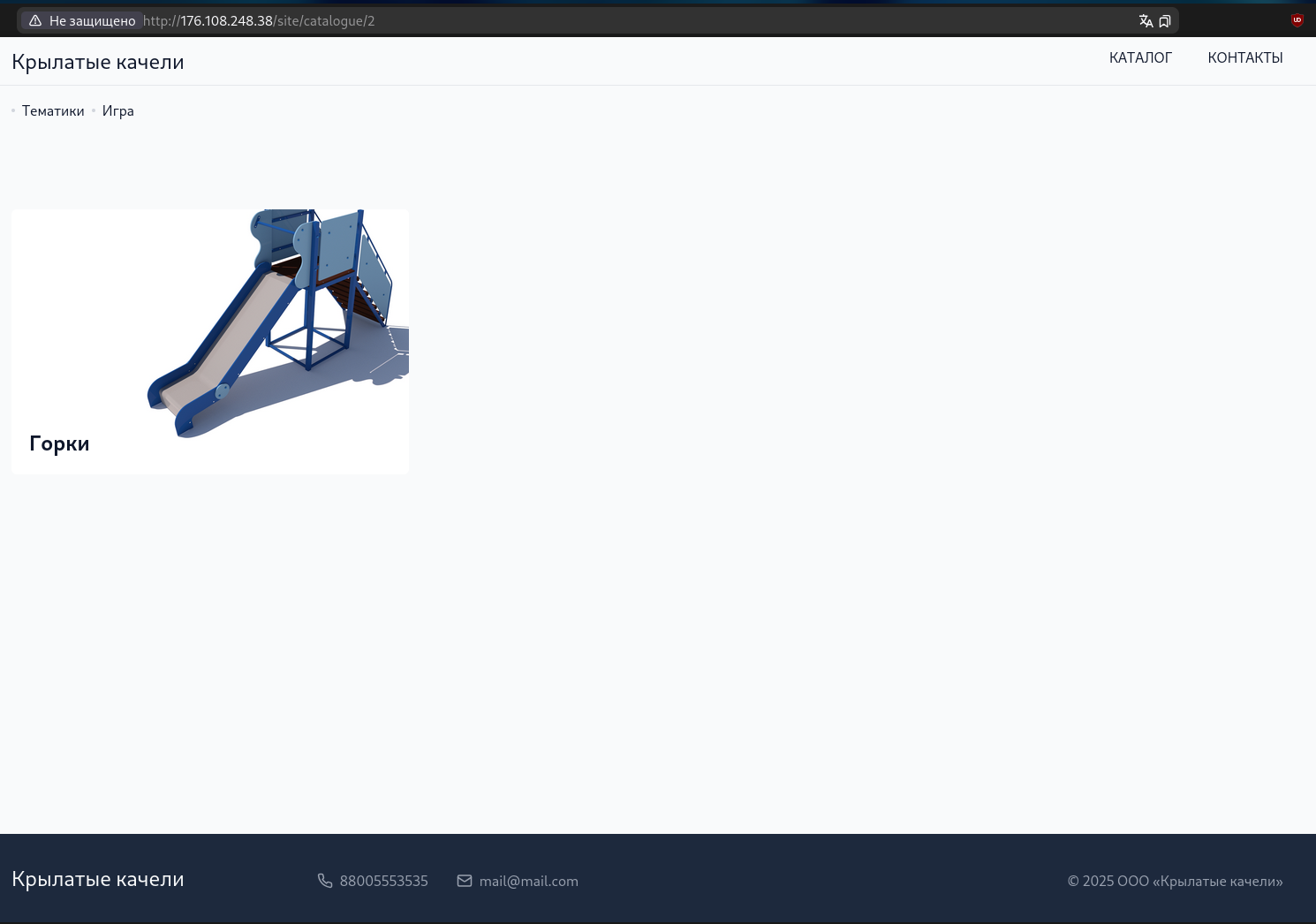


Рисунок 53 – Исправленный layout страницы

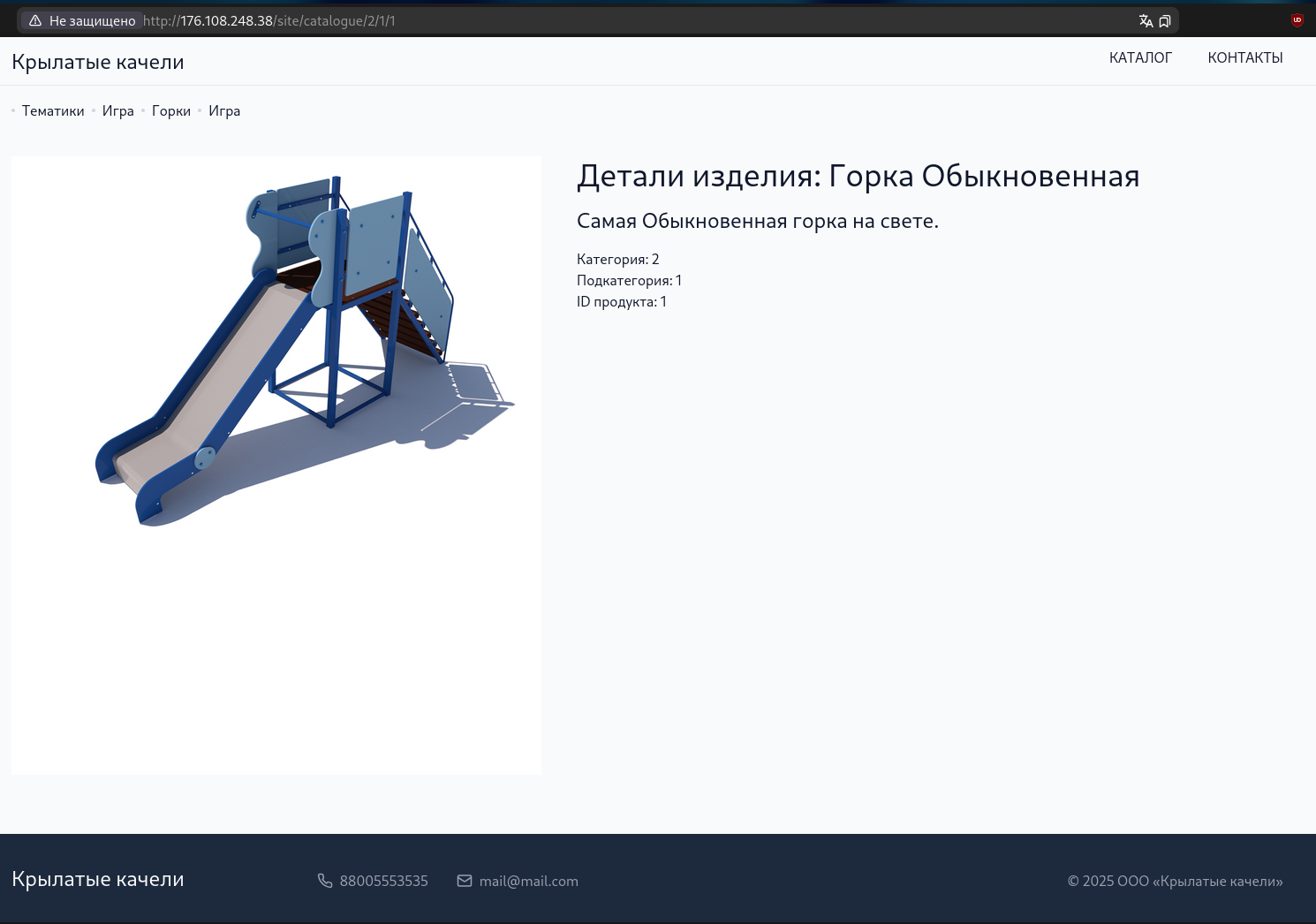


Рисунок 54 – Изменённая станица изделия

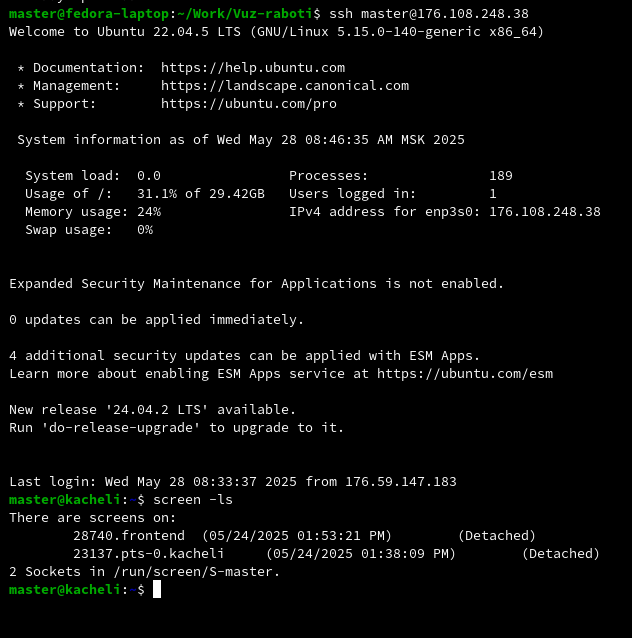


Рисунок 55 – Screens на виртуальной машине

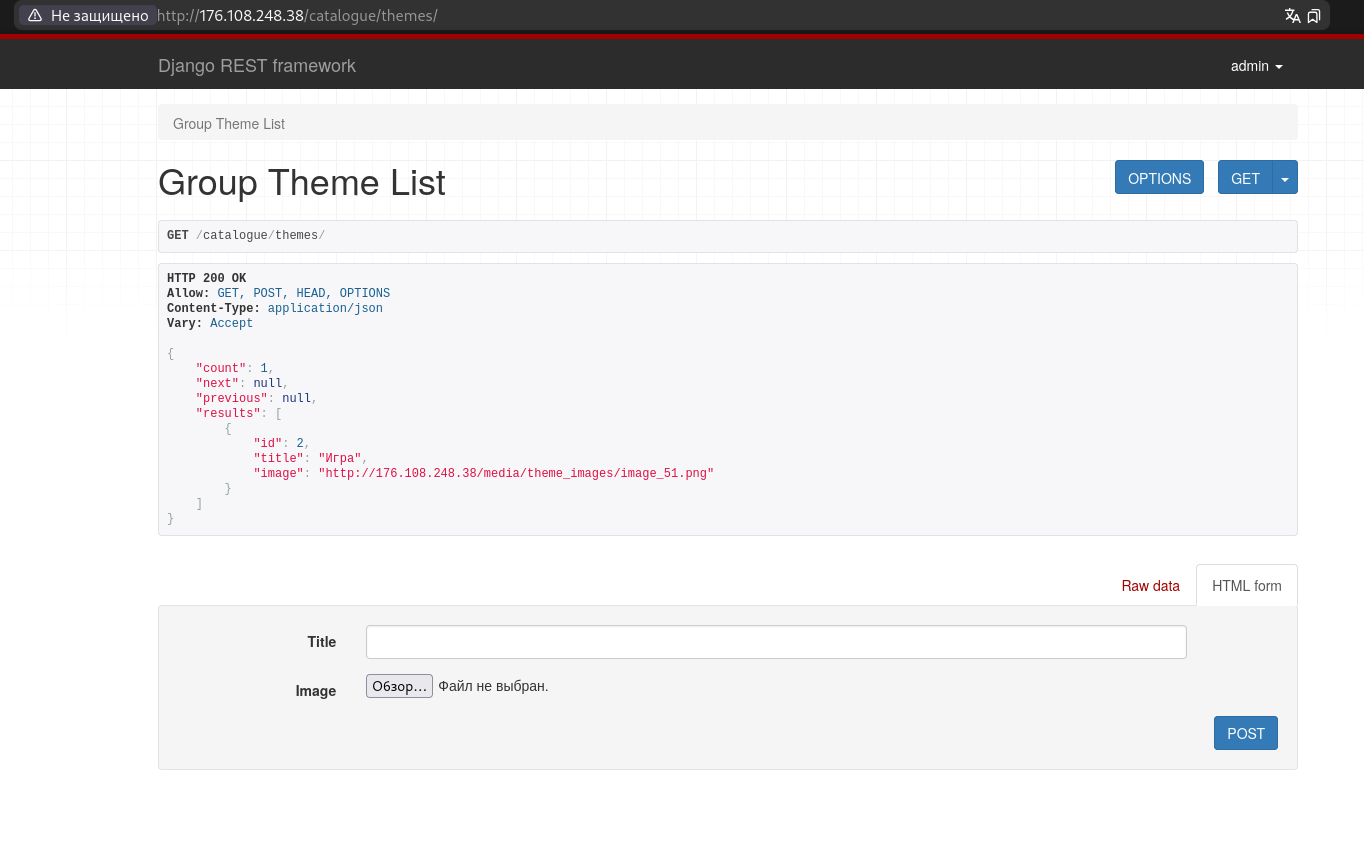


Рисунок 56 – Бэк на виртуальной машине



Рисунок 57 – Сайт на виртуальной машине.

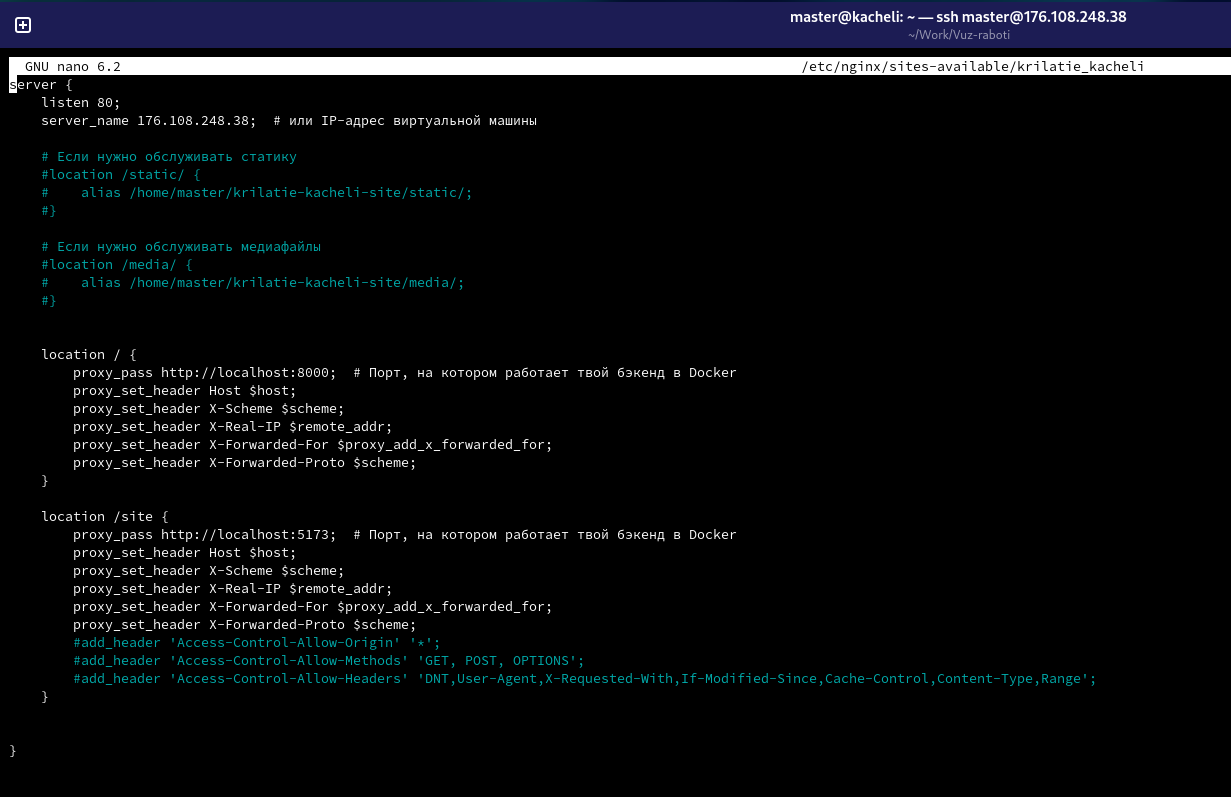


Рисунок 58 – Конфигурация Nginx

Я составил диаграмму сгорания задач (рисунок 7).

Рисунок 59 – Диаграмма сгорания задач

# Инструкция по использованию

Для того чтобы воспользоваться сайтом необходимо открыть любой веб-браузер и перейти на URL-адрес http://176.108.248.38/site. Можно посмотреть каталог и изделия в нём.

# ЗАКЛЮЧЕНИЕ

Разработка веб-приложения для работы с заметками позволила мне углубить знания в full-stack разработке и получить ценный практический опыт. В процессе работы я освоил создание устойчивой архитектуры, реализацию CRUD-функционала и интеграцию с API, а также научился писать unit-тесты.

Создание сайта-каталога позволило мне приобрести ценные навыки в веб-разработке и программировании, особенно в области дизайна и фронтенда. В процессе работы над проектом я получил опыт в full-stack разработке, а также научился эффективно взаимодействовать с бизнес-заказчиком. Работа с реальным коммерческим проектом стала важным этапом в моём профессиональном развитии, так как подобный опыт значительно повышает конкурентоспособность на рынке труда. Этот проект не только укрепил мои технические навыки, но и дал понимание ключевых аспектов деловой коммуникации в IT-сфере.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация Python [Электронный ресурс]. - Режим доступа: https://www.python.org/doc/ (Дата обращения: 4.06.2025)
2. Документация Django [Электронный ресурс]. - Режим доступа: https://docs.djangoproject.com/en/5.2/ (Дата обращения: 4.06.2025)
3. Документация Django REST Framework [Электронный ресурс]. - Режим доступа: https://www.django-rest-framework.org/tutorial/quickstart/ (Дата обращения: 4.06.2025)
4. Документация Typescript [Электронный ресурс]. - Режим доступа: https://www.typescriptlang.org/ (Дата обращения: 4.06.2025)
5. Документация React.js [Электронный ресурс]. - Режим доступа: https://react.dev/learn (Дата обращения: 4.06.2025)
6. Статья "Agile Software Development"[Электронный ресурс]. - Режим доступа: https://en.wikipedia.org/wiki/Agile\_software\_development (Дата обращения: 4.06.2025)
7. Документация Docker [Электронный ресурс]. - Режим доступа: https://docs.docker.com/ (Дата обращения: 4.06.2025)
8. Документация PostgreSQL [Электронный ресурс]. - Режим доступа: https://www.postgresql.org/docs/ (Дата обращения: 4.06.2025)
9. Документация Shadcnui [Электронный ресурс]. - Режим доступа: https://ui.shadcn.com/ (Дата обращения: 4.06.2025)
10. Документация Radix-ui [Электронный ресурс]. - Режим доступа: https://www.radix-ui.com/ (Дата обращения: 4.06.2025)
11. Документация pytest [Электронный ресурс]. - Режим доступа: https://docs.pytest.org/en/stable/ (Дата обращения: 4.06.2025)
12. Спецификация Swagger [Электронный ресурс]. - Режим доступа: https://docs.swagger.io/spec.html (Дата обращения: 4.06.2025)
13. Документация Nginx [Электронный ресурс]. - Режим доступа: https://nginx.org/en/docs/ (Дата обращения: 4.06.2025)
14. Документация Prometheus [Электронный ресурс]. - Режим доступа: https://prometheus-docs.netlify.app/docs/prometheus/latest/getting\_started/ (Дата обращения: 4.06.2025)
15. Документация Grafana [Электронный ресурс]. - Режим доступа: https://grafana.com/docs/ (Дата обращения: 4.06.2025)

# ПРИЛОЖЕНИЕ А

**Отрисовка UserFlow**

User Flow для сайта (рисунок А.1).



Рисунок А.1 – User Flow

# ПРИЛОЖЕНИЕ Б

**Диаграмма окон**

Диаграмма окон для сайта (рисунок Б.1).

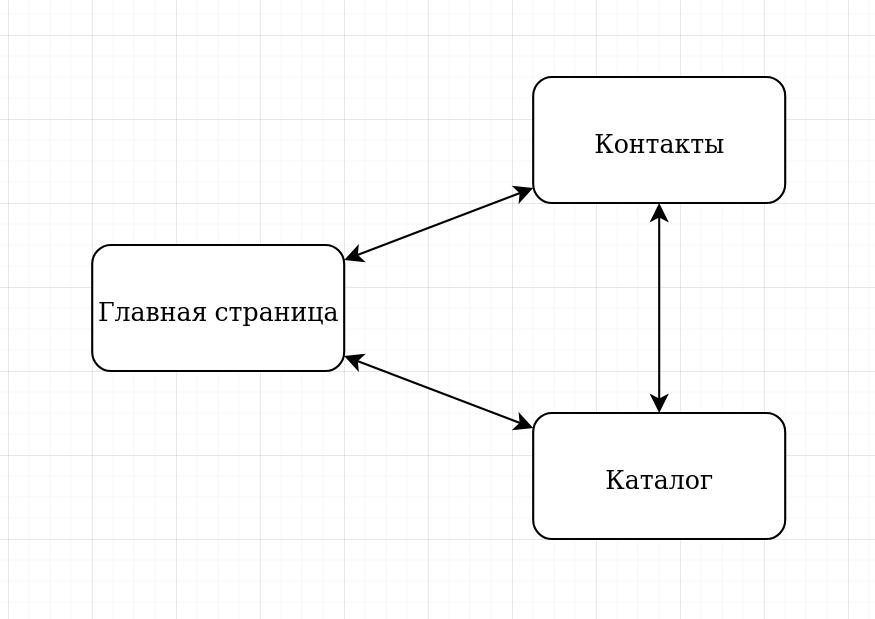


Рисунок Б.1 – Диаграмма окон

# ПРИЛОЖЕНИЕ В

**Диаграмма последовательности**

Диаграмма последовательности для сайта (рисунок В.1).

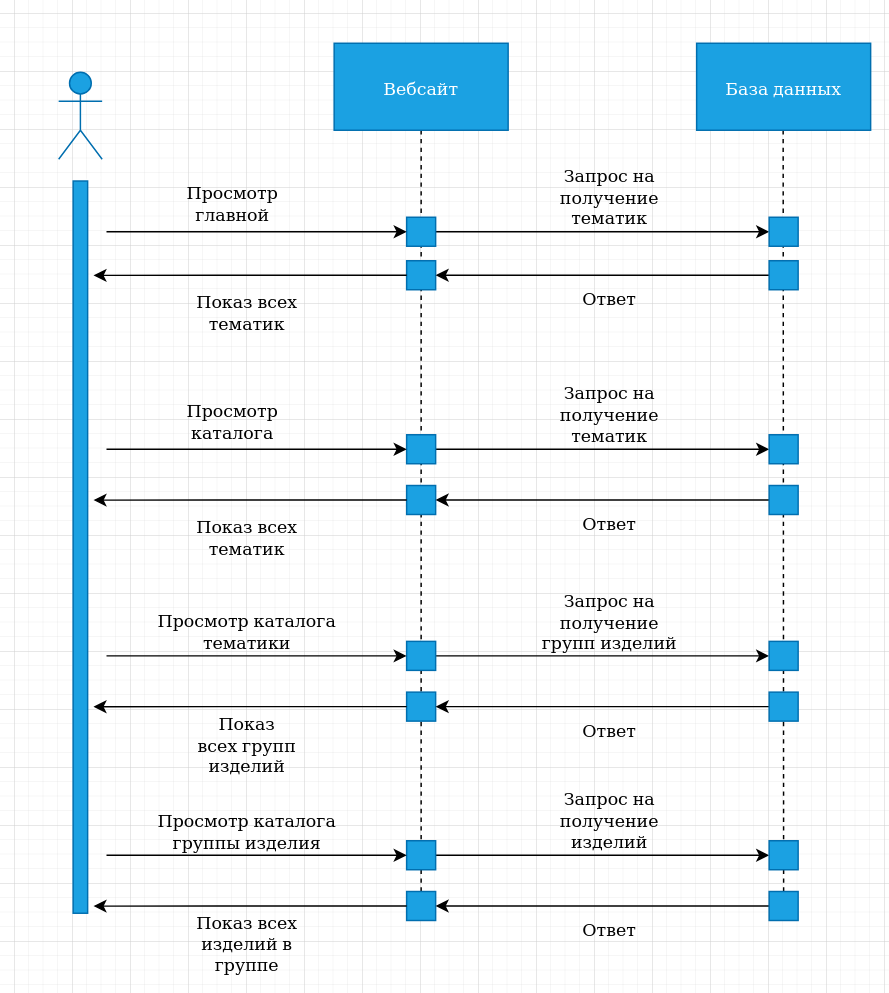


Рисунок В.1 – Диаграмма последовательности