

The background is a light blue surface covered with a repeating pattern of various tropical fruits and plants. The items include slices of watermelon with red flesh and green rinds, halves of papaya showing orange flesh and black seeds, whole and sliced lemons and limes, coconuts, and white plumeria flowers with green leaves. The central text is contained within a white, cloud-like shape.

Продуктовый Рай

Бережной Дмитрий и Тепляков Роман

The background features several abstract, organic shapes in a light pink color. Three slices of watermelon are scattered around: one in the top left, one in the bottom center, and one in the bottom right. Each slice shows a red interior with black seeds and a green rind.

01

Идея проекта

Идея проекта

Идея проекта заключается в том, чтобы создать сайт, который будет иммитировать работу продуктового магазина. Соответственно он включает все соответствующие элементы:

- Навигационную панель
- Формочки авторизации и регистрации
- Главную страницу с товарами по скидке и акциями
- Каталог с возможностью фильтрации товаров
- Страничку с информацией о конкретном товаре
- Личный кабинет с информацией о пользователе, возможностью её редактирования (через формочку) и всеми заказами пользователя
- Корзина пользователя, где отображаются все добавленные товары и имеется кнопка создания заказа (реализован именно процесс добавления объекта заказа в БД, механизма оплаты через карточку не реализовано, но в случае необходимости можно легко добавить в соответствующее api)
- Также имеется и админская страничка – здесь можно просмотреть всех пользователей (разумеется мы паролей не палим) и удалить в случае необходимости, редактировать и добавлять конечно же не можем :) Ещё есть вкладка с товарами – тут мы их можем уже не только удалять! Но и добавлять, а также редактировать (тоже через формочки). Аналогичная вкладка есть и для поставщиков



02

Реализация

Реализация

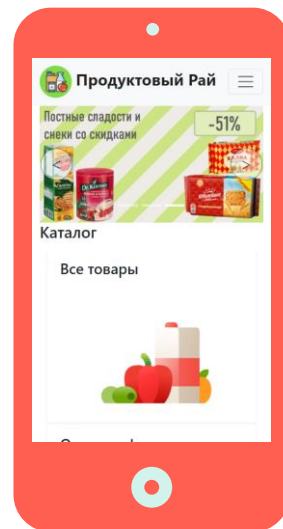
Проект написан на фреймворке Flask и всеми сопутствующими модулями. Взаимодействие с БД реализовано при помощи SQLAlchemy. Приложение имеет Flask-Restful архитектуру, т.е. в проекте много самодельных api – как ресурсов так блюпринтов. Используются они в случаях когда надо получить много какой-то информации или выполнить какие-то крупные вычисления, чтоб не засорять основной файл. Для api написаны автотесты на библиотеке pytest. Также в проекте много JS кода, который помогает обеспечивать работоспособность интерактивных элементов страницы: кнопочек для добавления товаров в корзину, слайдер на фильтре цены, кнопки перелистывания вкладок, всякие служебные элементы, по типу окошек с уведомлениями при удалении объектов на админской страничке и т.п. В frontend части приложения активно применялся bootstrap, но конечно же и самодельные стили никуда не делись – допустим те же карточки товаров имеют верстку с использованием css (bootstrap карточка сильно мелкая). Также используется и файлик с нормализацией стилей (взят с гитхаба), который помогает обеспечить одинаковую опорную точку для стилей абсолютно во всех браузерах (в разных браузерах могут отличаться отступы и верстка может поплыть). Касательно bootstrap – абсолютно все странички выстроены оп сетке => сайт можно открыть абсолютно со всех устройств и все будет красиво отображаться. Касательно обработки ошибок – она конечно же имеется, например, если попытаться удалить поставщика, который связан с каким-либо товаром, то ничего не выйдет - пользователь получит соответствующее уведомление и будет вынужден изменить поставщика в данных товарах перед тем, как удалить данного поставщика и подобных мелочей учтено много. Хостинг сайта был произведен с помощью сервиса Glitch

The background features several yellow organic shapes. There are four kiwi slices: two in the top left, one in the bottom center, and one in the bottom right. The kiwi slices are green with a yellow center and small brown seeds.

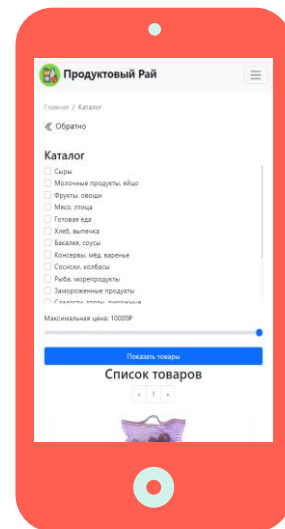
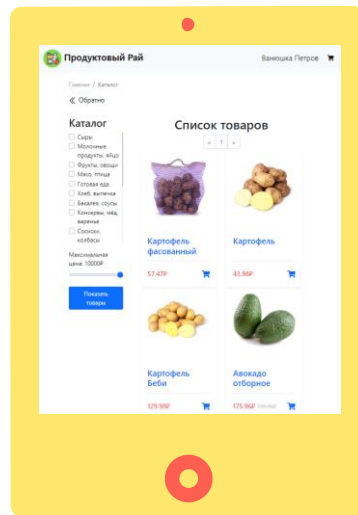
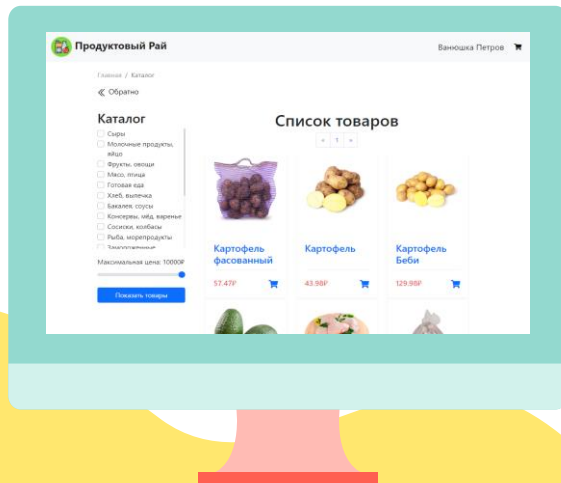
03

Краткий обзор

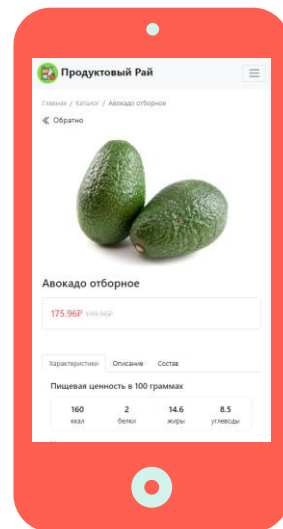
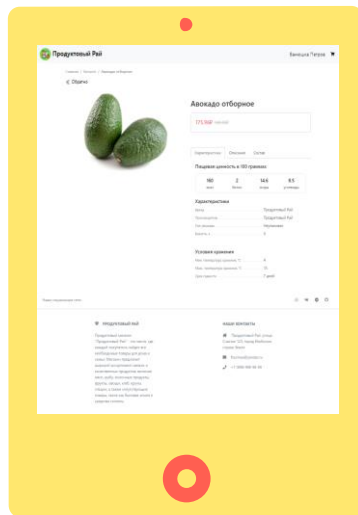
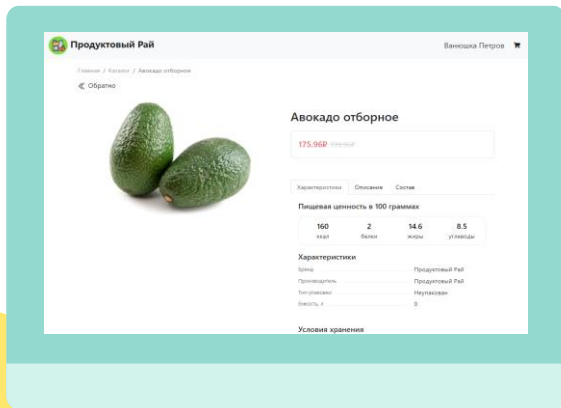
Главная страница



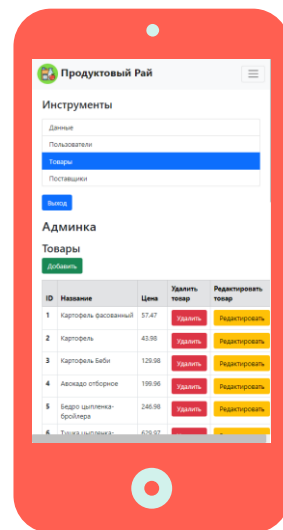
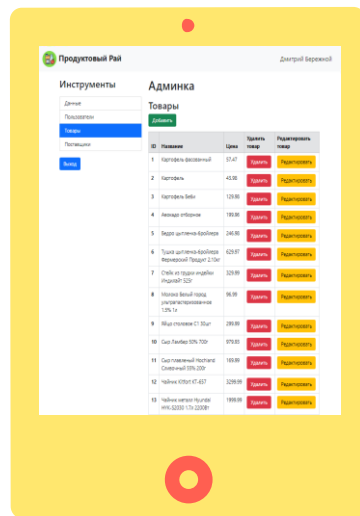
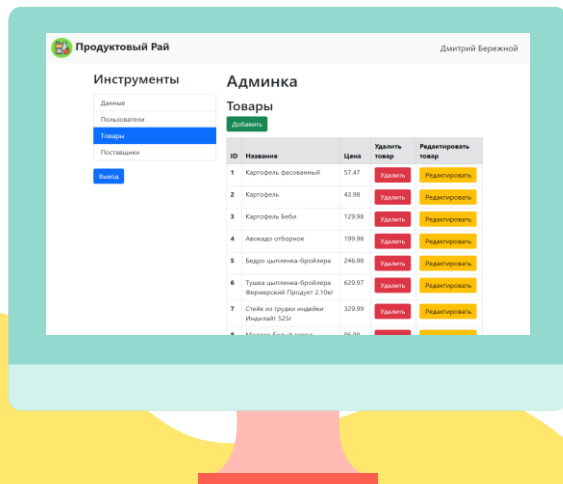
Каталог

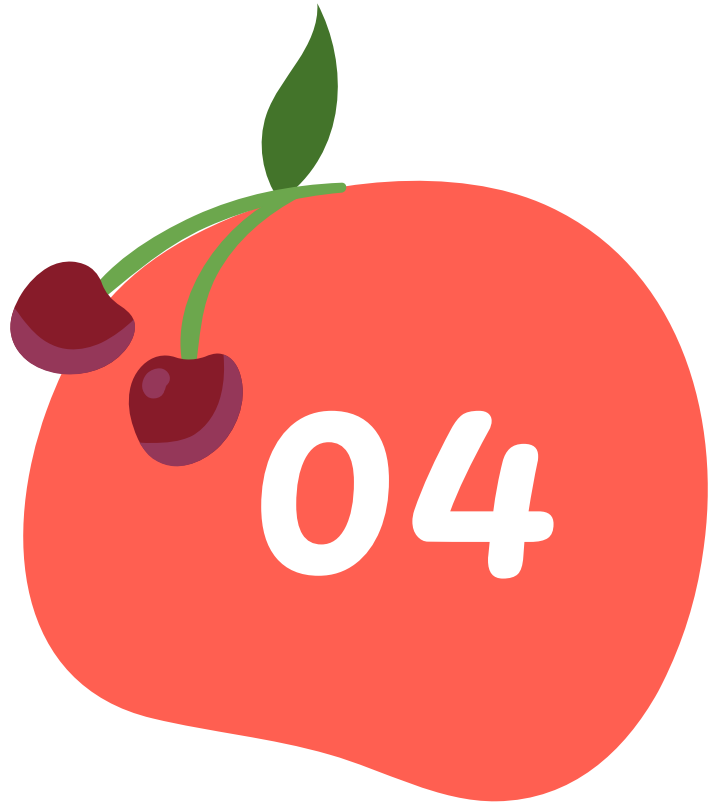


Информация о товаре

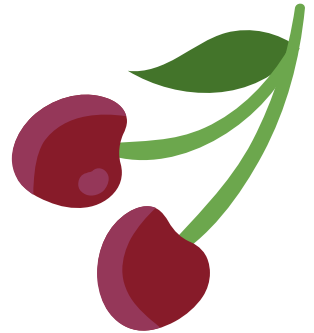
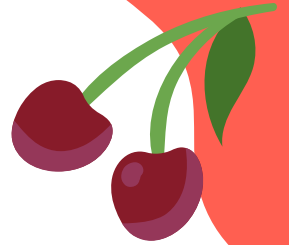


Админка





Umogu

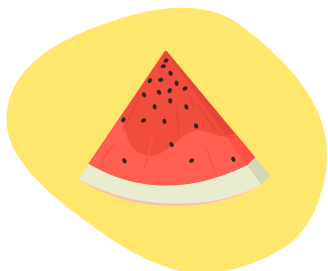


Статистика по строкам

Python: 2084 +

Python – слева кол-во строк непосредственно
самого приложения, справа – автотесты

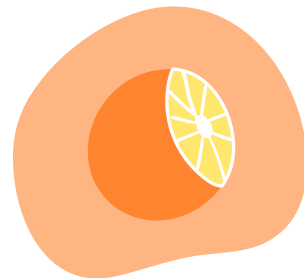
JS: 768



HTML: 1618



CSS: 432



Реализация

Проект написан на фреймворке Flask и всеми сопутствующими модулями. Взаимодействие с БД реализовано при помощи SQLAlchemy. Приложение имеет Flask-Restful архитектуру, т.е. в проекте много самодельных api – как ресурсов так блюпринтов. Используются они в случаях когда надо получить много какой-то информации или выполнить какие-то крупные вычисления, чтоб не засорять основной файл. Также в проекте много JS кода, который помогает обеспечивать работоспособность интерактивных элементов страницы: кнопочек для добавления товаров в корзину, слайдер на фильтре цены, кнопки перелистывания вкладок, всякие служебные элементы, по типу окошек с уведомлениями при удалении объектов на админской страничке и т.п. В frontend части приложения активно применялся bootstrap, но конечно же и самодельные стили никуда не делись – допустим те же карточки товаров имеют верстку с использованием css (bootstrap карточка сильно мелкая). Также используется и файл с нормализацией стилей (взят с гитхаба), который помогает обеспечить одинаковую опорную точку для стилей абсолютно во всех браузерах (в разных браузерах могут отличаться отступы и верстка может поплыть). Касательно bootstrap – абсолютно все странички выстроены оп сетке => сайт можно открыть абсолютно со всех устройств и все будет красиво отображаться. Касательно обработки ошибок – она конечно же имеется, например, если попытаться удалить поставщика, который связан с каким-либо товаром, то ничего не выйдет - пользователь получит соответствующее уведомление и будет вынужден изменить поставщика в данных товарах перед тем, как удалить данного поставщика и подобных мелочей учтено много. Хостинг сайта был произведен с помощью сервиса Glitch



Итоги

По итогу могу сказать, что получился очень масштабный проект (самый крупный за курс). В плане структуры, так же получилось наиболее удачное решение. И это был первый случай, когда разработка пошла ровно по плану и каких-либо непредвиденных обстоятельств не возникло. Можно сказать, что проект получился безупречным :)



Спасибо за
внимание!

