Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»

Образовательная программа «Веб-технологии»

Отчет по курсовому проекту

по дисциплине «Инженерное проектирование»

Тема: «Анкетирование пользователей сайта»

**Выполнил:**

Студент группы 191-321

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Огородников Н. А.

подпись, дата

**Принял:**

Старший преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Даньшина М.В.

подпись, дата

Москва 2021

Оглавление

[Оглавление 2](#_Toc77325653)

[Введение 3](#_Toc77325654)

[Тип разработки 3](#_Toc77325655)

[Задача 3](#_Toc77325656)

[Аналоги 3](#_Toc77325657)

[Анализ аналогов 4](#_Toc77325658)

[Проектирование 5](#_Toc77325659)

[Кейсы использования 5](#_Toc77325660)

[Выбор функционала 5](#_Toc77325661)

[Разработка и тестирование 6](#_Toc77325662)

[Архитектура проекта 6](#_Toc77325663)

[Структура базы данных 6](#_Toc77325664)

[Типовые запросы к базе данных 7](#_Toc77325665)

[Программирование 8](#_Toc77325666)

[Дизайн 9](#_Toc77325667)

[Тестирование 11](#_Toc77325668)

[Заключение 13](#_Toc77325669)

[Список используемых источников 14](#_Toc77325670)

Введение

Тип разработки

Индивидуальный проект, направленный на демонстрацию и фиксацию навыков, полученных в течение 3-го семестра обучения. Разработка ведется поэтапно.

Задача

Спроектировать и создать реляционную базу данных для веб-сервиса анкетирования пользователей сайта и настроить административную панель для управления базой данных с помощью фреймворка Django, а также создать веб-сайт анкетирования, использующий данную базу данных. Сайт панель поможет создавать анкеты и вопросы, и просматривать ответы.

Аналоги

* [Google Forms](https://www.google.com/intl/ru_ua/forms/about/)
* [Survio](https://www.survio.com/ru/)
* [EXAMINARE](https://www.examinare.ru/)
* [Simpoll](https://simpoll.ru/)
* [Анкетолог](https://anketolog.ru/)

Анализ аналогов

Сервисы анкетирования широко распространены в наше время, поэтому у разрабатываемого решения есть множество аналогов. Вот некоторые из них:

* [Google Forms](https://www.google.com/intl/ru_ua/forms/about/)
* [Survio](https://www.survio.com/ru/)
* [EXAMINARE](https://www.examinare.ru/)
* [Simpoll](https://simpoll.ru/)
* [Анкетолог](https://anketolog.ru/)

Данные сервисы имеют схожую структуру, и процесс создания анкеты в них сводится к трем основным шагам:

1. Создание анкеты
2. Добавление секций анкеты (минимум 1)
3. Добавление вопросов (минимум 1)

Данные сервисы предоставляют широкие возможности по настройке анкет: добавление описаний, настройка типов вопросов, распределение на логические элементы (секции).

Все сервисы позволяют просматривать ответы, предоставленные пользователями, с целью дальнейшего анализа. Также есть поддержка экспорта ответов в большинстве современных форматов.

Опыт использования сервисов положительный. Интерфейс удобный и интуитивно понятный, присутствует адаптивность, что позволяет использовать сервисы на мобильных устройствах.

Подводя итог можно сказать, что сайты являются типовыми решениями, и имеют между собой высокую конкуренцию. Данные сервисы можно использовать как ориентир при разработке проекта.

Проектирование

Кейсы использования

Можно выделить 4 роли пользователей сервиса:

* Администратор – имеет полный доступ к базе данных и всем разделам сервиса, единственный, кто имеет доступ к административному интерфейсу
* Оператор – может смотреть анкеты всех пользователей и ответы к ним, но не может изменять или удалять, имеет доступ к дашборду со статистикой
* Пользователь – может создавать изменять свои анкеты, управлять доступом к ним (открывать/закрывать для прохождения) и просматривать ответы на них, не имеет доступ к статистике
* Опрашиваемый – незарегистрированный в системе пользователь, который может отвечать на открытые анкеты

Выбор функционала

На основе кейсов использования, сайт должен иметь следующий функционал:

* Административный интерфейс для Администратора
* Личный кабинет для Администратора, Оператора и Пользователя
* Создание, чтение, изменение и удаление анкет
* Создание и чтение ответов на анкеты
* Страница для прохождения анкеты, не требующая авторизации
* Страница со статистикой (дашборд)

Разработка и тестирование

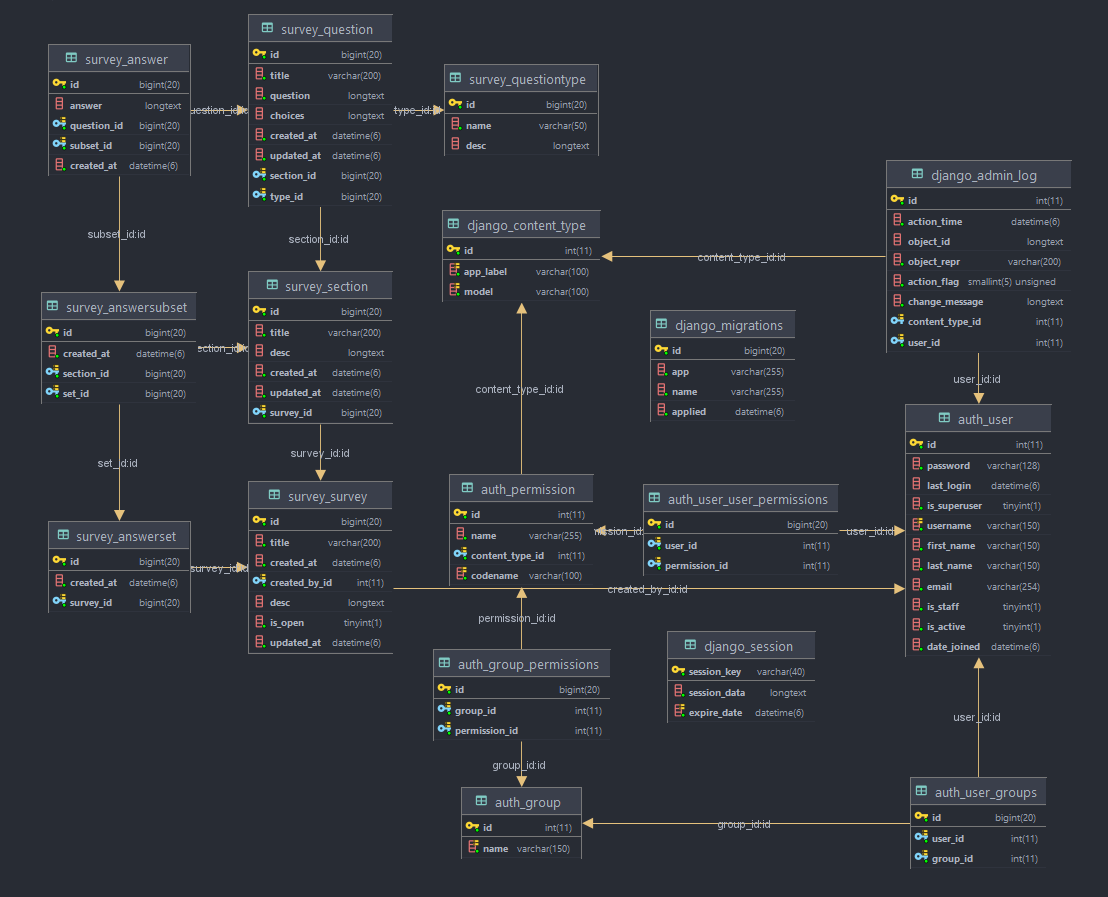
Архитектура проекта

Проект состоят из базы данных MySql и приложения Django, ответственного за взаимодействие с БД и отображения веб-страниц

Структура базы данных

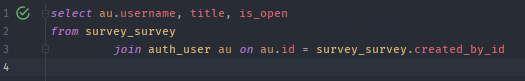
Следуя из анализа конкурентов, сервис должен строится на двух основных процессах: создание анкет и анализ ответов. Анкета состоит из секций, которые состоят из вопросов. Вопрос может быть определенного типа (один из, несколько из, ответ в свободной форме и).

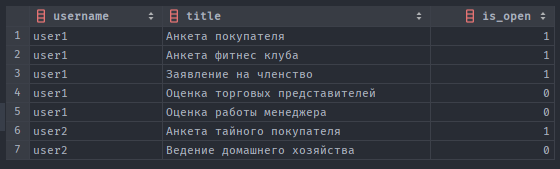
Ответы на анкеты хранятся похожим образом: Набор ответов на анкету ссылается на соответствующую анкету, набор ответов на секцию ссылается на соответствующую сукцию и на набор ответов на анкету, ответ хранит в себе ответ и ссылается на соответствующий вопрос и набор ответов на секцию.



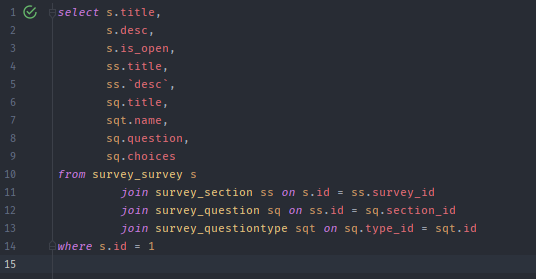
Типовые запросы к базе данных

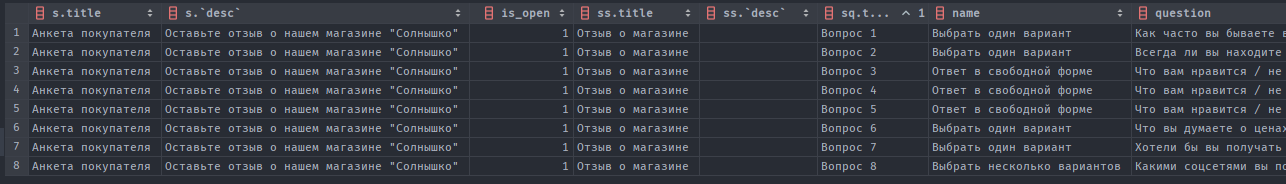
Запрос на вывод всех анкет пользователей:



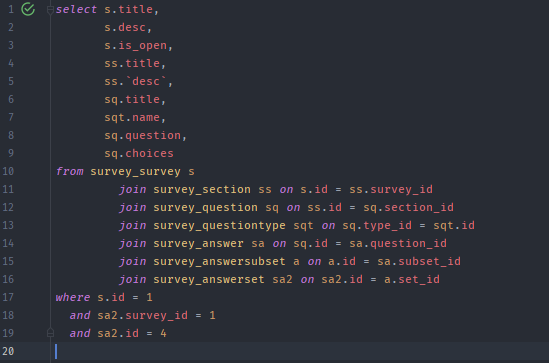


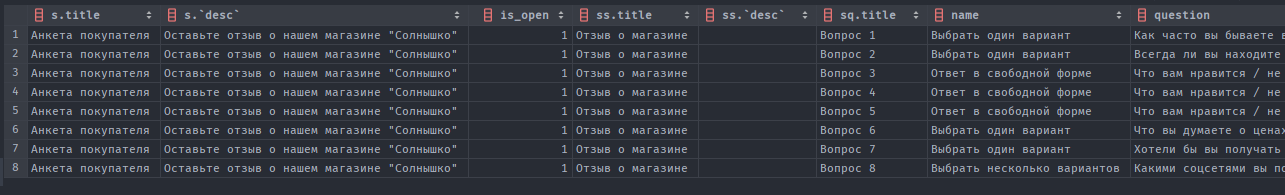
Запрос на вывод анкеты:



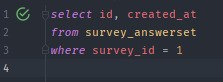


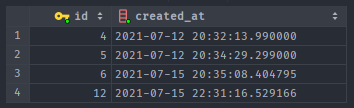
Запрос на вывод ответов на анкету:





Запрос на вывод списка ответов на анкету:





Программирование

Приложение работает по следующим принципам:

* Фреймоворк Django осуществляет взаимодействие с базой данных
* Формирование контента сайта происходит динамически на стороне сервера
* Отображение веб страниц осуществляется с помощью Django Templates

В приложении реализован CRUD. Сайт может:

* Создавать анкеты, секции, вопросы, ответы
* Выводить информацию для чтения (ответы)
* Обновлять поля (анкеты, секции, вопросы)
* Удалять анкеты, секции, вопросы

В приложении реализована фильтрация, но только в административном интерфейсе.

В приложении реализован поиск. В личном кабинете можно искать анкеты по названию, а в случае для администратора и оператора поиск будет осуществлять также и по именам пользователей.

В приложении реализованы роли, они были описаны в главе Проектирование

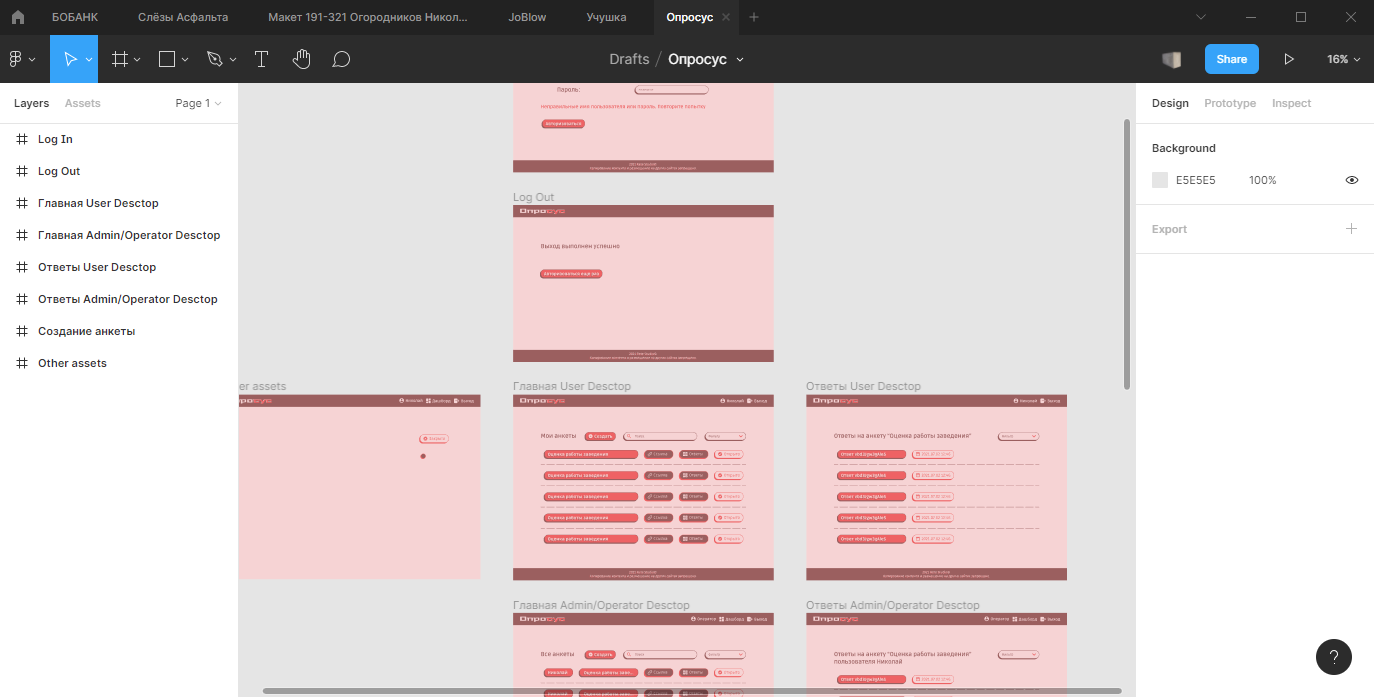
В приложении реализован дашборд

В приложении уделено внимание безопасности. Посетив серию вебинаров по компьютерной безопасности, я извлек для себя много полезной информации и понял значимость высоких стандартов безопасности. Поэтому в приложении реализована авторизация через внутренний модуль фреймворка. Он хранит пароли в зашифрованном виде что повышает безопасность

Дизайн

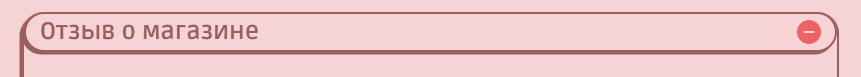
Данное приложение имеет современный дизайн, выполненный в духе минимализма. При создании дизайна использовались самые современные решения в данной области:

* Использование Figma. Прототип дизайна приложения был разработан с помощью данного инструмента. И пусть итоговый вариант сайта выглядит иначе, он все равно продолжает и развивает идеи, заложенные в прототипе

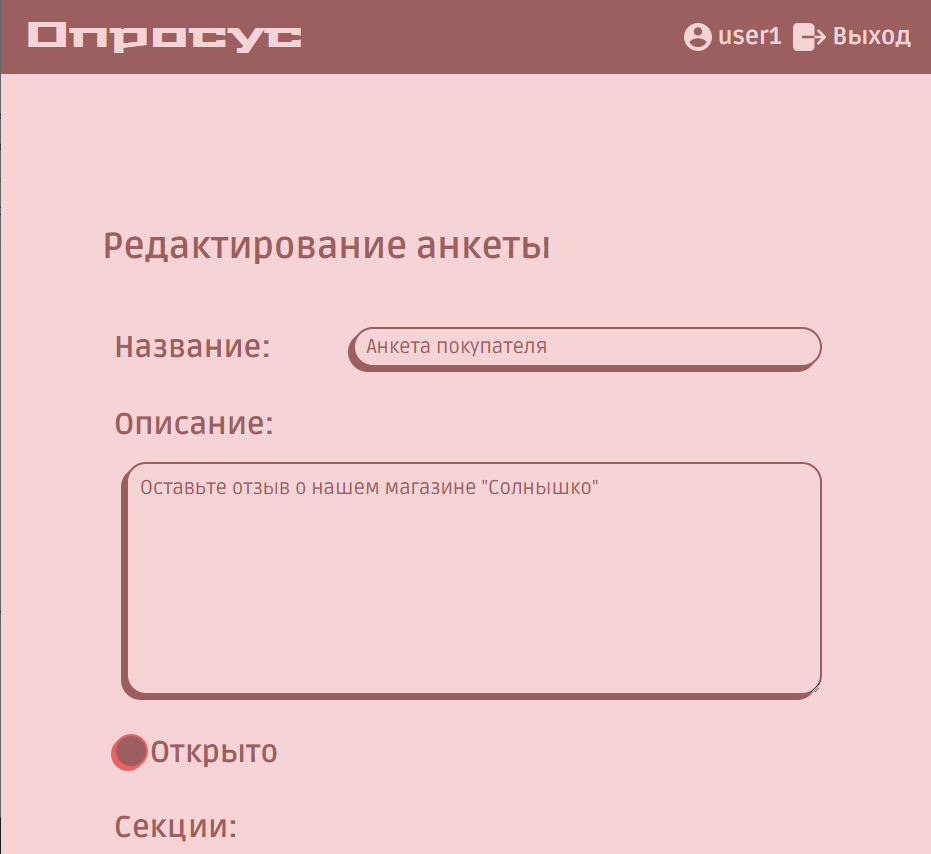


* Использование SVG иконок. Иконки с обладают отличной узнаваемостью и помогают в создании интуитивно понятного пользовательского интерфейса, а формат SVG позволяет управлять видом иконки (менять цвет, размер и т.д.)

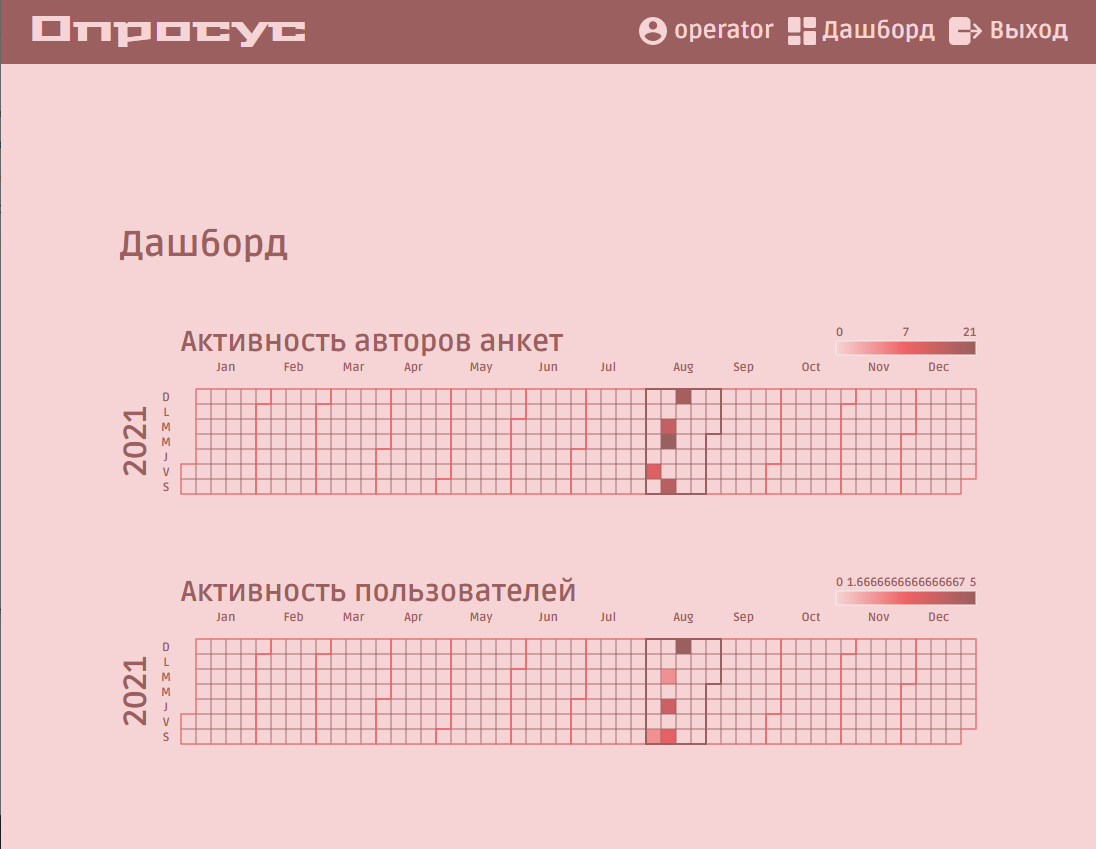




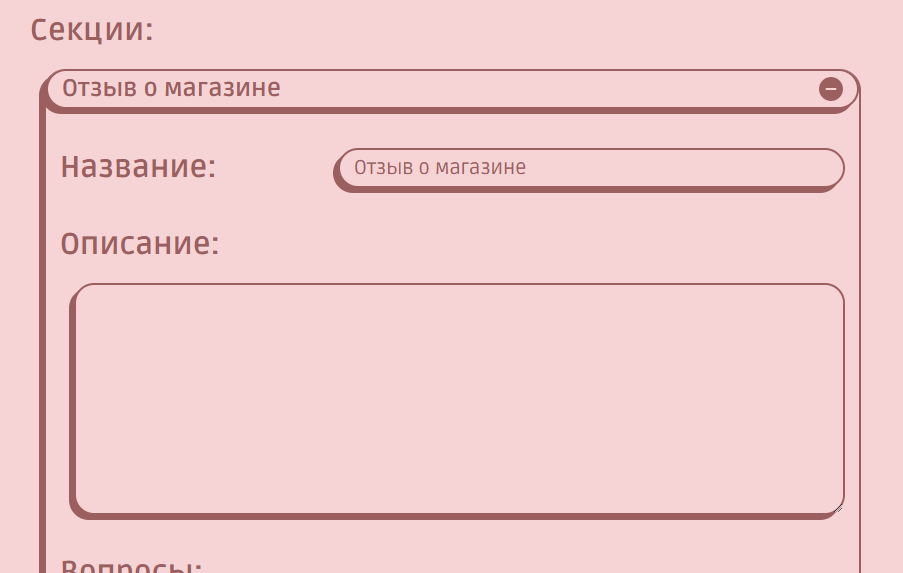
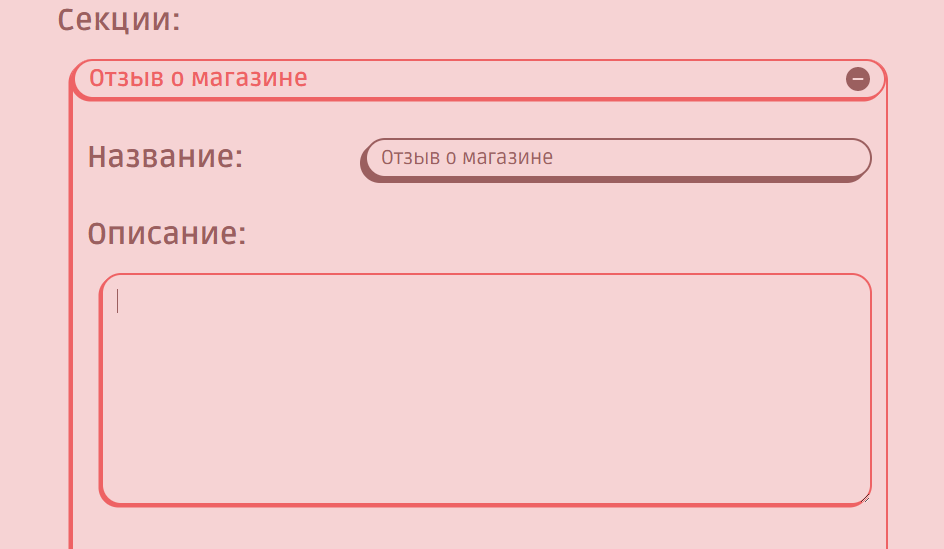
* Стилизация выделения текста. Позволяет избавиться от синенго выделения и вписать его в дизайн страницы

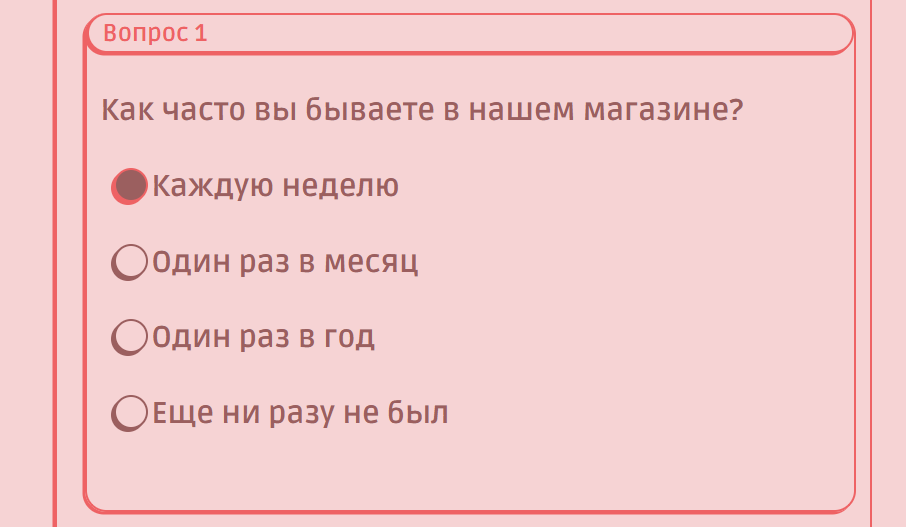
* Использование Google Charts. Данная библиотека позволила создать на странице диаграммы активности пользователей и стилизовать их под дизайн сайта



* Использование тени для создания эффекта неравномерной обводки. Изменение размера тени в сочетании с hover и transition позволяет обозначить интерактивные элементы в статичные элементы

* Стилевое оформление checkbox. Прописать стили конкретно данному элементу не выйдет, но можно сделать свой checkbox.



* Указание размера шрифтов в rem позволяет повысить доступность сайта.

Тестирование

Для данного приложения проводилось ручное тестирование с имитацией поведения обычного пользователя.

В ходе проведения тестирования были выявлены и исправлены следующие проблемы:

* Повторная отправка POST данных при перезагрузке страницы
* Проблемы с навигацией
* Проблема защиты данных
* Проблема, при которой на странице не появлялись наборы ответов на анкету
* Проблемы с некоторыми запросами в БД
* Проблема с обработкой url фреймворком

По полученным результатам можно сделать вывод, что технология ручного тестирования в данном случае показала себя с наилучшей стороны и помогла выявить ряд фатальных проблем продукта.

Заключение

В результате процесса разработки получилось то, что и изначально планировалось: веб-сайт для анкетирования.

Также в процессе создания данного приложения я применил знания и навыки, полученные за прошлые курсы обучения (python-разработка, проектирование баз данных, html, css, sass, js), освежил их и даже в некоторой степени развил.

Полноценная Full-Stack разработка была интересным опытом в моей практике, и я намерен дальше совершенствовать свои навыки в данном направлении.

Git-репозиторий: <https://github.com/kolaqsq/Site_users_survey>

Макет: <https://www.figma.com/file/FMaqsPvL5U4otPkjbkzzA2/%D0%9E%D0%BF%D1%80%D0%BE%D1%81%D1%83%D1%81?node-id=0%3A1>

Приложение: <http://oprosus.std-950.ist.mospolytech.ru/>

Административный интерфейс: <http://oprosus.std-950.ist.mospolytech.ru/admin/>

Данные пользователей (логин/пароль):

* admin/admin
* operator/operator
* user1/user1
* user2/user2

Список используемых источников

<https://docs.djangoproject.com/en/3.1/>

<https://webdevblog.ru/kak-prevratit-adminku-django-v-legkovesnuju-panel-instrumentov>

<https://djbook.ru/>

<https://django-import-export.readthedocs.io/en/latest/>

<https://django.fun/tutorials/put-ot-request-do-response-v-django/>

<https://pythonist.ru/kastomizacziya-admin-paneli-django/>

<https://developers.google.com/chart>