

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО”

Факультет ИКТ

Образовательная программа Интеллектуальные системы в гуманитарной сфере

Направление подготовки (специальность) Интеллектуальные системы в гуманитарной
сфере (45.03.04)

О Т Ч Е Т

по курсовой работе

Тема задания: РЕАЛИЗАЦИЯ WEB-СЕРВИСОВ СРЕДСТВАМИ Django REST framework.

Обучающийся: Цыбаева Арина Олеговна К3343

Руководитель: Говоров А. И.

Подписи членов комиссии:

Говоров А.И.

(подпись)

Чунаев А.В.

(подпись)

Антонов М.Б.

(подпись)

Дата _____

Санкт-Петербург
2020

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ	2
1. ПРОЕКТИРОВАНИЕ	3
1.1 Проектирование web-сервиса	4
1.2. Проектирование базы данных	6
2. РЕАЛИЗАЦИЯ WEB-СЕРВИСА	8
2.1. Общая информация	8
2.2. Реализация интерфейса “Главная страница”	9
2.3. Реализация интерфейса “Сделать заказ”	11
2.4. Реализация интерфейса “Наши водители”	13
2.5. Реализация интерфейса “Склад”	15
2.6. Реализация интерфейса “Отчет”	18
2.7. Реализация интерфейсов “Входа”, “Выхода”, “Регистрации” пользователей	20
ЗАКЛЮЧЕНИЕ	22
СПИСОК ЛИТЕРАТУРЫ	23

ВВЕДЕНИЕ

В настоящее время тема экологии начинает волновать все больше и больше людей, поэтому открывается множество сервисов для помощи утилизации вторичного сырья или же сбора мусора в борьбе со свалками и мусоросжигательными полигонами. Сервис Экотакси является одним из них. Его функционал состоит в том, чтобы забрать отходы у клиента и впоследствии сдать на мусороперерабатывающий завод. Данный сервис освобождает клиента от поиска места сдачи определенной категории мусора, транспортировки желаемого сдать на переработку мусора, тем самым предоставляя клиенту больше свободного времени для нерутинной жизни.

Цель курсовой работы заключалась в реализации web-сервиса ЭкоТакси, которое могла бы использовать схожая компания для сбора отчетности, приема заказов и отслеживания работы сотрудников.

Средствами Django Rest Framework был реализован бэкенд web-сервиса, Vue.js был выбран для реализации фронтенда, а также использована база данных - PostgreSQL.

Задачи курсовой работы:

- Выявление функциональных требований.
- Реализация функциональных требований.
- Проектирование дизайна веб-сервиса.
- Написание отчета о проделанной работе.

1. ПРОЕКТИРОВАНИЕ

1.1. Проектирование web-сервиса

Для реализации web-сервиса ЭкоТакси за основу был взят Django Rest Framework. С помощью него реализовано сервисное отображение данных, выборка необходимой информации, добавление новой информации и исполнение запросов в соответствии с поставленными задачами функционала.

Vue.js с плагином Vuetify использовались для осуществления фронтенда приложения. Vuetify был выбран из соображений быстрой скорости сборки клиентской части, а также удобства и простоты дизайна. Представления web-сервиса отображены в папке Components и полностью удовлетворяют запрашиваемому функционалу.

В качестве базы данных была использована база данных PostgreSQL, настройка базы данных, а также последующая работа и сборка приложения производилась с помощью редактора PyCharm. Для удобства запуска, а также быстроты дальнейшей сборки на желаемом сервисе были добавлены Docker контейнеры для каждой части архитектуры приложения.

Так, диаграмму архитектуры разрабатываемого web-сервиса представлены на Рисунке 1.

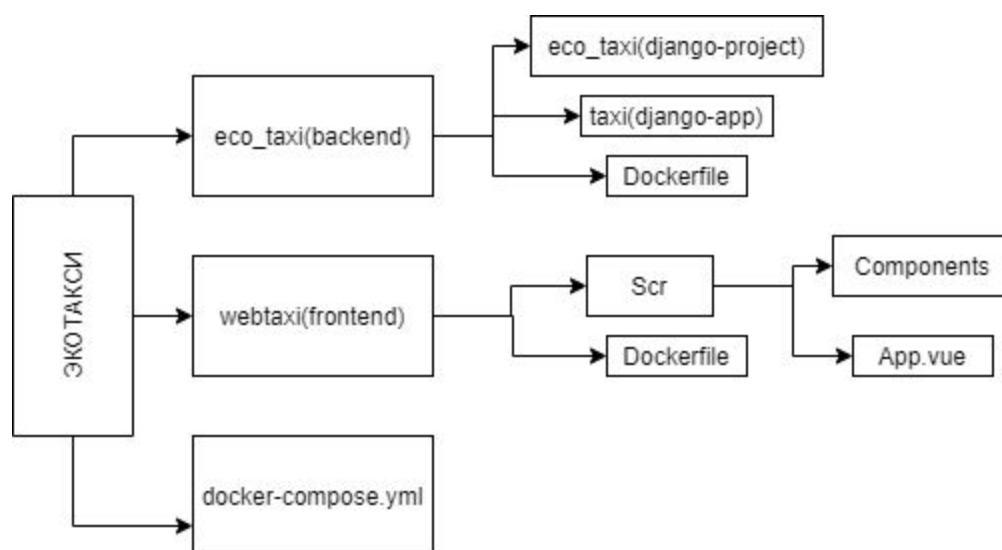


Рисунок 1 - Архитектура web-сервиса ЭкоТакси

Задачи реализации функционала web-сервиса:

- Отображение списка клиентов моложе 40 - молодые активисты, использующие сервис.
- Отображение востребованности сервиса в процентном соотношении в зависимости от дня недели.
- Отображение топ-3 сдаваемых категории мусора за все время работы сервиса.
- Отображение количества(кг) каждой категории на складе.
- Отображение заказов определенным водителем за определенную дату.
- Добавление нового заказа.
- Предоставление администратору прав доступа для найма/увольнения водителя; добавления заказа на склад, списание заказа со склада.
- Предоставление администратору прав доступа для просмотра отчета за текущий месяц.

1.2. Проектирование базы данных

В качестве сущностей для разрабатываемого web-сервиса были выбраны Клиент, Водитель, Категория мусора и склад. Связывающая сущность - Заказ. Сущность Склад добавлена, чтобы выводить отчетность за месяц: необходимо определять, сколько мусора сдано на переработку на заводы утилизации отходов. Для этого администратору необходимо принимать заказ на склад (после того, как водитель забрал его у клиента), а также отмечать, какие заказы он сдал на склад (количество мусора в кг подсчитывается автоматически).

Реализация вышеописанной базы данных представлена на Рисунке 2. Желтым цветом помечены первичные ключи сущностей, синим - внешние ключи сущностей.

Представленные сущности:

1. Клиент: сбор данных о клиенте - ФИО, возраст, Адрес, Email, Телефон.
2. Заказ: информация о сделанном заказе - Исполняющий водитель, Клиент, Категория сдаваемого мусора, Стоимость, Масса и Дата исполнения.
3. Водитель: сбор данных о водителе - ФИО, Фотография, Возраст, Email, Телефон, Марка и Модель Машины, Номер Машины.
4. Категория мусора: реализуемые на сдачу категории мусора - Название категории и описание.
5. Склад: реализация добавления и списания со склада заказов; хранится только id заказов.

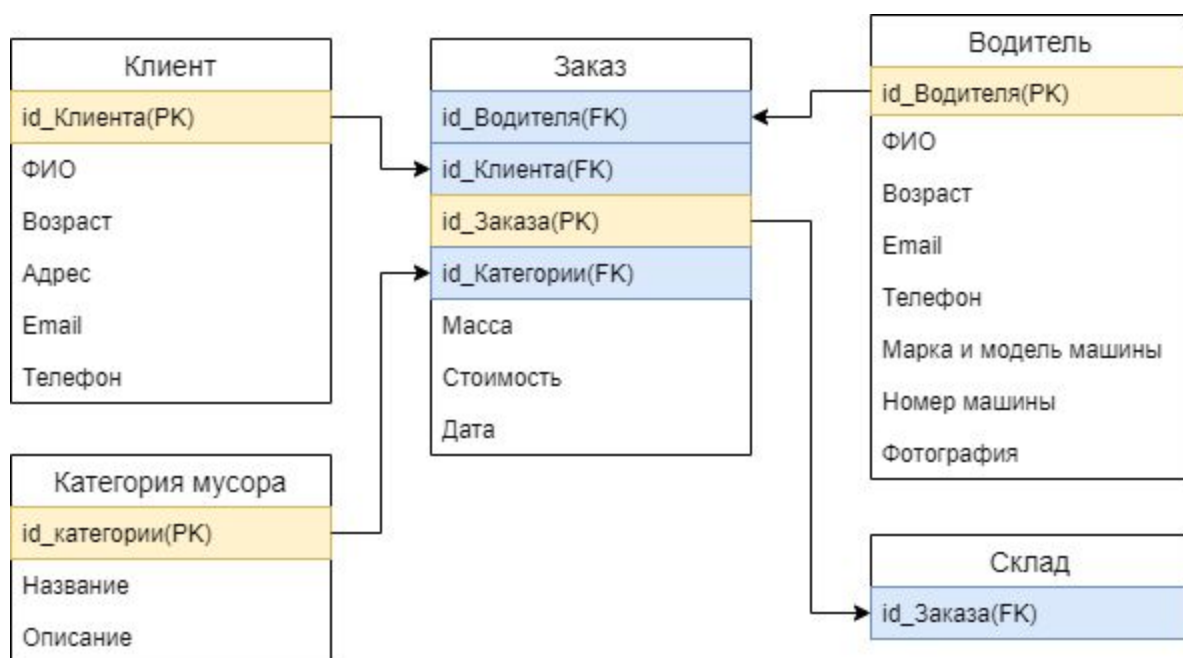


Рисунок 2 - Диаграмма базы данных web-сервиса

2. РЕАЛИЗАЦИЯ WEB-СЕРВИСА

2.1. Общая информация

Проект разделен на следующие главные интерфейсы: Главная страница, Добавление заказа, Водители, Склад, Отчет. Также существуют кнопки для входа/выхода и регистрации пользователя.

Вся навигация в проекте, впрочем, как и дизайн реализован с помощью плагина Vue.js Vuetify. Существует Navigation-drawer, с помощью которого пользователь может перемещаться по web-сервису(Рисунок 3).

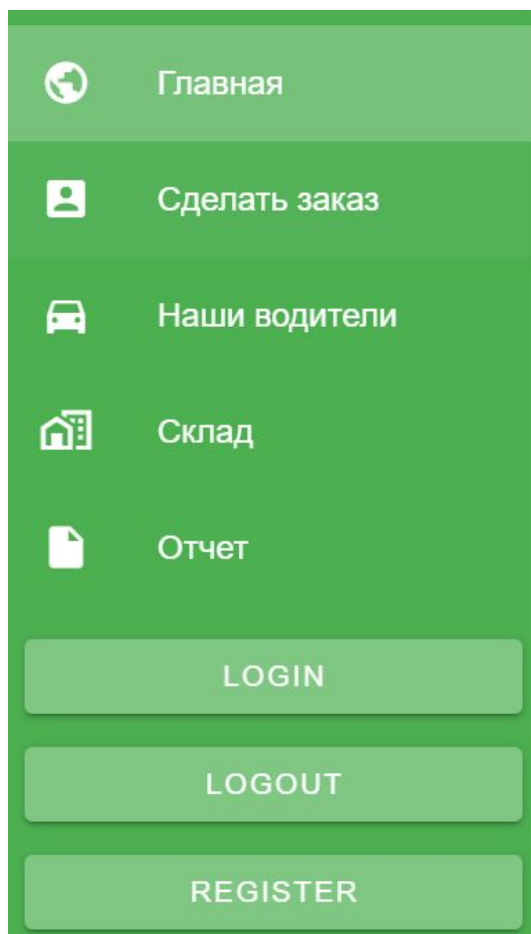


Рисунок 3 - Navigation-drawer web-сервиса.

2.2. Реализация интерфейса “Главная страница”

Данный интерфейс отображает общую информацию о пользователях, соответствующая поставленным задачам. На главной странице реализованы три запроса: вывод списка клиентов моложе 40(1), вывод процентной востребованности сервисом в каждый день недели(2), а также топ-3 сдаваемых категории мусора(3). Сортировка и выборка необходимых данных происходит на бэкенде, получаемые данные отображаются с помощью таблицы v-simple-table.

1) Запрос о выводе списка клиентов моложе 40 лет:

Данная задача реализована с помощью модуля django-filters. Был создан фильтр-ClientFilter, который сортирует клиентов по возрасту. Далее с помощью сериалайзера, а также библиотеки generics.ListAPIView выводится список отсортированных клиентов. Сортировка происходит в самом запросе путем добавления к ссылке аргумента “age_max” и приравнивания аргумента к нужному значению. Пример ссылки: 'http://127.0.0.1:8000/client/?age_max=40', реализованный вывод списка клиентов представлен на Рисунке 4.

Наши молодые активисты:

Фамилия	Возраст
Цыбаева Арина Олеговна	20
Куликов Артем Васильевич	40
Чужакова	32
ofis	23
Alisa	9
Луковина Анастасия	22
Арсенюк	20
Пупкин Василий	20

Рисунок 4 - Реализация запроса вывода списка клиентов моложе 40 лет

2) Запрос о выводе процентного соотношения востребованности web-сервиса в каждый день недели.

Реализация данной задачи производилась также с помощью библиотеки generics.ListAPIView и функции get_queryset(). Бралось общее количество заказов, сгруппированных по дню недели, считалось их количество по id, делилось на общее количество заказов и умножалось на 100%. Данное значение записывалось новым столбцом

“percent” и выводилось с помощью сериалайзера в таблицу на клиентской части (Рисунок 5): цифры соответствуют дням недели (отсчет ведется с воскресенья).

Количество заказов в зависимости от дня недели:

День недели	Процентная занятость
1	7 %
2	23 %
3	3 %
4	34 %
5	11 %
6	7 %
7	11 %

Рисунок 5 - Реализация вывода востребованности web-сервиса в зависимости от дня недели

3) Запрос на вывод топ-3 сдаваемых категории мусора:

Реализация данной задачи производилась также с помощью библиотеки `generics.ListAPIView` и функции `get_queryset()` [1]. За основу бралась таблица “Заказ”, данные группировались по значению “category__name”, сортировались по убыванию по количеству мусора с помощью функции `order_by()`, затем по индексу выбиралось 3 первых элемента. Так как имя категории - это внешний ключ таблицы “Заказ”, для корректного и читаемого отображения было необходимо добавить в сериалайзер источник данных: `(source='category__name')`. Реализация представленной задачи отображена на Рисунке 6.

Топ-3 самых сдаваемых категорий мусора:

Компост

Стекло

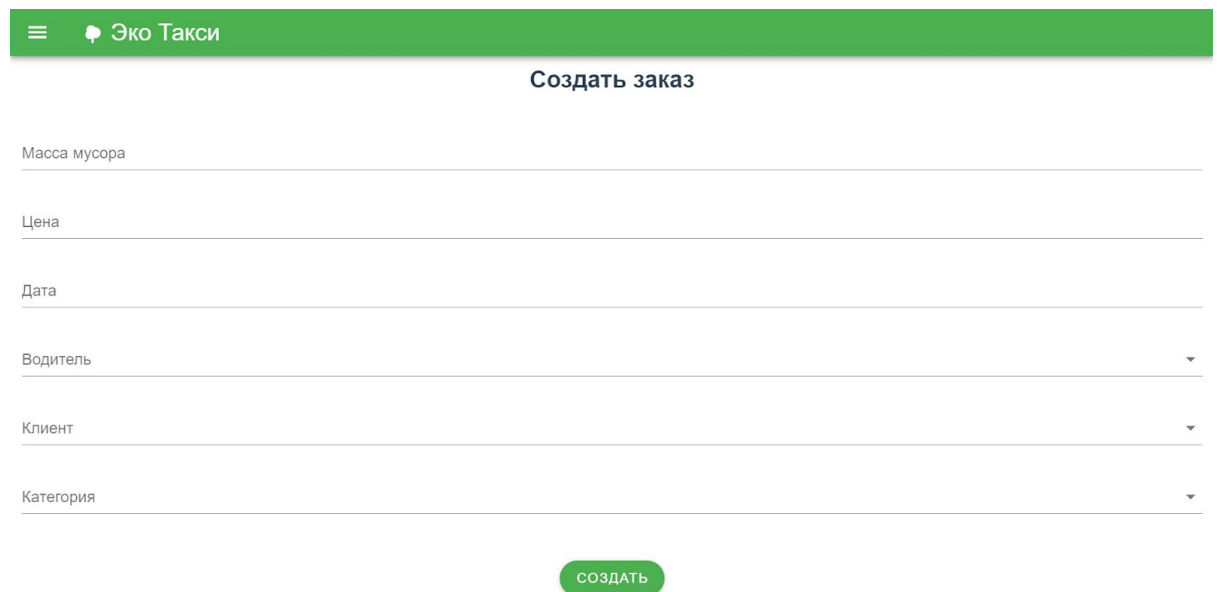
Твердый пластик

Рисунок 6 - Реализация вывода списка топ-3 сдаваемые категории мусора

2.3. Реализация интерфейса “Сделать заказ”

Реализация данного интерфейса производилась с помощью библиотеки Djano Rest Framework - `generics.CreateAPIView`. На представленной странице пользователю предоставляется возможность создать заказ - создать новую запись в базе данных в таблице “Заказ”. Для этого были созданы (аналогично `generics.ListAPIView`) `serializer`, а также `django view` с определением прав доступа (с помощью библиотеки `permissions`) - заказ может создать только авторизованный пользователь.

Так как Водитель, Клиент и Категория мусора являются внешними ключами в сущности “Заказ” возникла сложность при вводе данных. Необходимо было реализовать ввод данных именно названиями, а не их `id`. Проблема была решена на уровне фронтенда - данные, получаемые с бэкенда были преобразованы в словарь, где `text` - это необходимое название, а `value` - `id`, которое уже непосредственно и записывается в форму. Далее было принято решение сделать выпадающие списки для удобства и быстроты заполнения. Выпадающие списки сделаны с помощью `vueify - v-select` [4, 6]. Пример реализации интерфейса представлен на Рисунке 7, пример выпадающего списка можно увидеть на Рисунке 8.



Эко Такси

Создать заказ

Масса мусора

Цена

Дата

Водитель

Клиент

Категория

СОЗДАТЬ

Рисунок 7 - Реализация интерфейса “Создать Заказ”

Водитель

Маркелов
Завьялова
Новикова
Пупкин

Рисунок 8 - Реализация выпадающего списка

При успешном создании заказа появляется выпадающее окно с надписью “Спасибо за заказ!” - Рисунок 9.

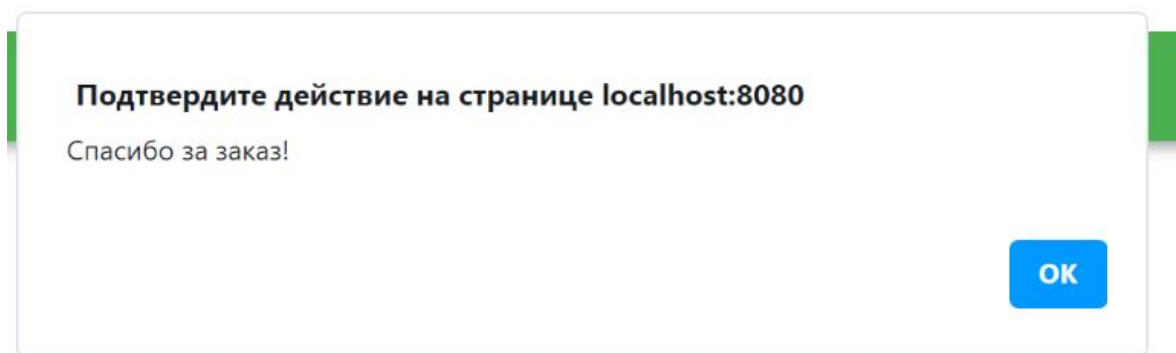
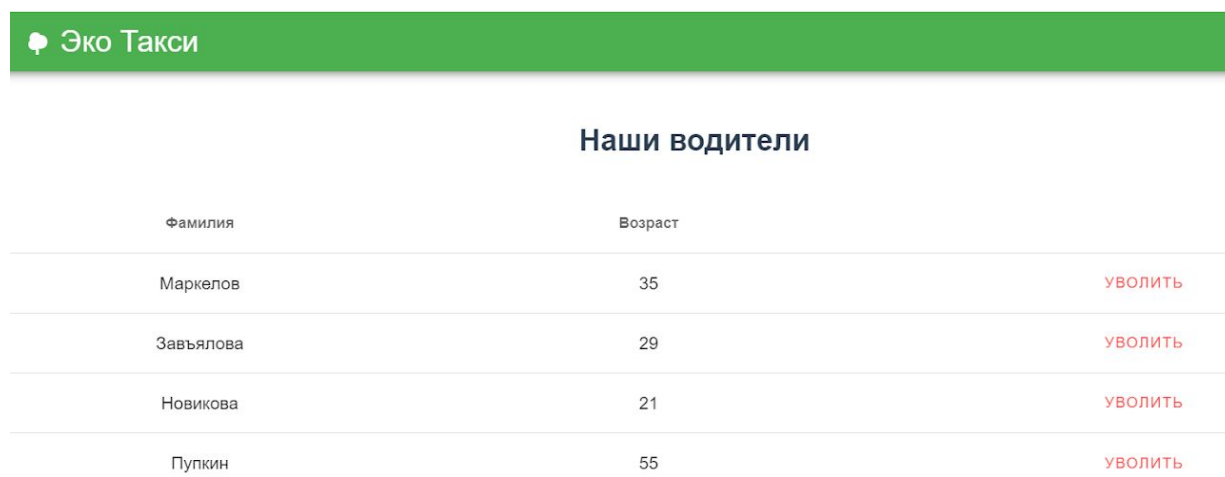


Рисунок 9 - Всплывающее окно, подтверждающее успешное создание заказа

2.4. Реализация интерфейса “Наши водители”

В данном интерфейсе важную роль играют права доступа. Администратор может просмотреть список водителей, нанять и уволить водителя, в то время как пользователь может только посмотреть список водителей. Данные возможности предоставляются с помощью библиотеки permissions. Добавление водителя реализовано аналогично добавлению заказа. Удаление (увольнение) реализовано при помощи библиотеки Django Rest Framework viewsets.ModelViewSet, также в файл `eco_taxi/urls.py` добавлен словарь с ключами : `get`, `put`, `patch`, `delete` - и соответствующими значениями - `retrieve`, `update`, `partial_update`, `destroy` [1]. На Рисунках 10, 11 представлен вышеописанный интерфейс (с помощью кнопки “УВОЛИТЬ” происходит увольнение выбранного водителя).



Эко Такси		
Наши водители		
Фамилия	Возраст	
Маркелов	35	УВОЛИТЬ
Завьялова	29	УВОЛИТЬ
Новикова	21	УВОЛИТЬ
Пупкин	55	УВОЛИТЬ

Рисунок 10 - Список водителей с кнопкой “УВОЛИТЬ”

☰

Эко Такси

Добавить водителя

Фамилия

Возраст

Email

Телефон

Модель и марка машины

Номер машины

ДОБАВИТЬ

Рисунок 11 - Форма добавления нового водителя

2.5. Реализация интерфейса “Склад”

Данный интерфейс предназначен только для сотрудников штата/администрации, поэтому также было необходимо обеспечить корректные права доступа для использования функционала. Права доступа также, как и в других ранее описанных интерфейсах, предоставляются `django.permissions`. На данной странице производится реализация запроса количество (кг) каждого вида мусора, хранящегося на складе(то есть забранного у клиента, но еще не сданного на мусороперерабатывающий завод). Данные группируются по названию категории из таблицы “Заказ” и считается общая масса каждой категории. Данные отображаются с помощью библиотеки Django Rest Framework `generics.ListAPIView` и представлены в качестве таблицы `vuetify v-simple-table` [5]. Также для удобства выведена справка по наименованиям и названиям категорий. Реализация вышеупомянутого интерфейса представлена на Рисунке 12.

Категория	Килограммы
3	20
5	8
1	5

Наименование	Название категории
1	Компост
2	Твердый пластик
3	Мякий пластик
4	Макулатура
5	Стекло
6	Тетра-Пак
7	Металл
8	Лампочки и батарейки

Рисунок 12 - Реализация интерфейса “Склад”

Также страница содержит две кнопки: “Принять заказ”(1) и “Списать заказ”(2) для добавления и списания заказа соответственно.

1) Кнопка “Принять заказ”.

Кнопка “Принять заказ” перенаправляет пользователя на страницу принятия заказа. С точки зрения базы данных Администратор из таблицы “Заказ” добавляет записи в таблицу “Склад”. Реализация страницы “Добавление заказа на склад” представлена на Рисунке 13.

Добавить заказ на склад

Номер заказа

ДОБАВИТЬ

ВЕРНУТЬСЯ НА СКЛАД

Номер заказа	Категория	Масса	Дата	Водитель
141	Стекло	2	2020-05-25	Маркелов
142	Макулатура	12	2020-05-25	Маркелов
143	Мягкий пластик	30	2020-05-25	Маркелов
144	Мягкий пластик	20	2020-05-25	Маркелов
145	Твердый пластик	200	2020-05-25	Завьялова
146	Твердый пластик	200	2020-06-15	Новикова
10	Компост	5	2020-05-17	Завьялова

Рисунок 13 - Реализация страницы “Добавить заказ на склад”

На странице “Добавить заказ на склад” выводится список всех созданных заказов; чтобы добавить заказ на склад, необходимо ввести номер заказа в окно “Номер заказа” и нажать кнопку “Добавить”. Список заказов выведен с помощью библиотеки Django Rest Framework generics.ListAPIView, добавление заказа на склад происходит по id. Доступ к этой странице разрешен только сотрудникам штата и/или администратору. Также для удобства добавлена кнопка “Вернуться на склад”, которая перенаправляет пользователя на страницу “Склад”.

2) Кнопка “Списать заказ”.

Кнопка “Списать заказ” перенаправляет пользователя на страницу списания заказа со склада (предполагается, что данный интерфейс будет необходим в случае сдачи принятого у клиентов мусора на завод по утилизации отходов). Реализация страницы “Списать заказ” представлена на Рисунке 14.

Заказы списываются при помощи кнопки “Списать” напротив каждого заказа. Реализована данная таблица заказов аналогично таблице работающих в компании водителей.

Данные сортируются и выводятся с помощью библиотеки Django Rest Framework. Также для удобства добавлена кнопка “Вернуться на склад”, которая перенаправляет пользователя на интерфейс “Склад”.

ВЕРНУТЬСЯ НА СКЛАД

Списать заказ со склада

Выберите заказ, который хотите списать:

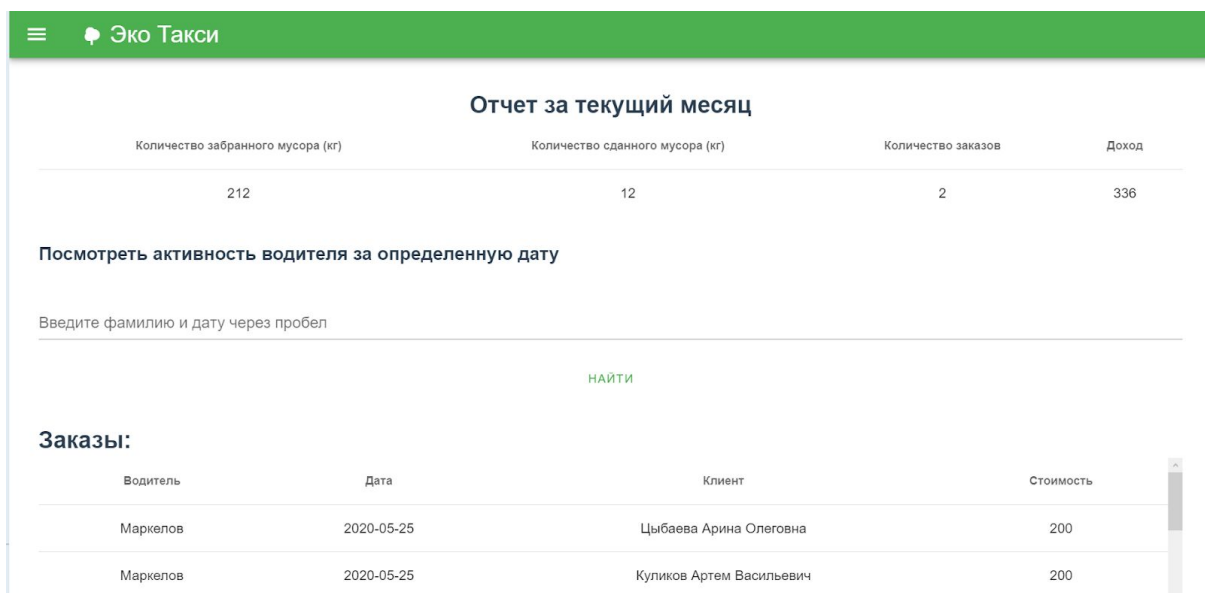
Номер заказа	
1	СПИСАТЬ
10	СПИСАТЬ
11	СПИСАТЬ
12	СПИСАТЬ
12	СПИСАТЬ
143	СПИСАТЬ
144	СПИСАТЬ

Рисунок 14 - Реализация страницы “Списать заказ со склада”

2.6. Реализация интерфейса “Отчет”

По данному техническому заданию веб-сервис должен содержать отчет за каждый месяц с количеством забранного и сданного мусора, количеством заказов и доходом. Нахождение количества собранного мусора осуществлялось с помощью фильтрации по дате записей из сущности “Заказ”, сложения массы всех заказов за месяц. Количество сданного мусора подсчитывалось через сущность “Склад”: из массы всех заказов за месяц вычиталась сумма заказов, оставшихся на складе. Количество заказов и доход компании за месяц подсчитывался аналогично сумме всех заказов за месяц по группировке данных из сущности “Заказ” за текущий месяц. Также установлены права доступа только для сотрудников штата/администрации. Реализацию вышеописанного интерфейса можно увидеть на Рисунке 15.

На той же странице располагается поиск заказов по фамилии водителей, а также по дате исполнения заказа. Реализация поиска происходит с помощью библиотеки Django Rest Framework filters класса SearchFilter по полям ‘data’, ‘driver__name’ [7]. Для поиска необходимо нажать на кнопку “Найти”. Поиск без введенных данных также представлен на Рисунке 15. Поиск с введенными данными изображен на Рисунках 16, 17.



Отчет за текущий месяц			
Количество забранного мусора (кг)	Количество сданного мусора (кг)	Количество заказов	Доход
212	12	2	336

Посмотреть активность водителя за определенную дату

Введите фамилию и дату через пробел

НАЙТИ

Заказы:

Водитель	Дата	Клиент	Стоимость
Маркелов	2020-05-25	Цыбаева Арина Олеговна	200
Маркелов	2020-05-25	Куликов Артем Васильевич	200

Рисунок 15 - Реализация интерфейса “Отчет”

Посмотреть активность водителя за определенную дату

Введите фамилию и дату через пробел
Маркелов

НАЙТИ

Заказы:

Водитель	Дата	Клиент	Стоимость
Маркелов	2019-10-24	Вальтов Добрыня Ильич	4
Маркелов	2020-06-12	Куликов Артем Васильевич	36
Маркелов	2020-05-25	Куликов Артем Васильевич	100

Рисунок 16 - Реализация поиска по фамилии водителя.

Посмотреть активность водителя за определенную дату

Введите фамилию и дату через пробел
Завъялова 2020-05-14

НАЙТИ

Заказы:

Водитель	Дата	Клиент	Стоимость
Завъялова	2020-05-14	Куликов Артем Васильевич	200

Рисунок 17 - Реализация поиска по фамилии водителя и дате исполнения заказа.

2.7. Реализация интерфейсов “Входа”, “Выхода”, “Регистрации” пользователей

Реализация данных интерфейсов производится с помощью Djoser, Simple JWT, DRF auth token [2]. Для удобства выполнения работы в начале были реализованы запросы в приложении Postman для понимания структуры исполнения запросов, а также для проверки вывода запрашиваемой информации из базы данных. Далее те же запросы, что были выполнены в приложении Postman, были прописаны в Vue.js файлах, добавлено всплывающее окно при успешном входе в систему, а также вывод ошибки при неправильном вводе данных [3].

Продумана логика событий регистрации: вначале пользователь придумывает логин и пароль, затем заполняет информацию о себе (сущность “Клиент”), и в конце авторизируется, как пользователь и может приступить к нужным действиям. Регистрация клиента, добавление дополнительной информации о нём представлены на Рисунках 18, 19.

Страница Входа и Выхода из приложения исполнена достаточно просто. На страница Выхода пользователю предоставляется возможность войти заново. Вход и Выход реализованы на Рисунках 20, 21.

Авторизация

Логин

Пароль

ВОЙТИ

Вы вышли из системы!

ВОЙТИ

Рисунок 18, 19 - Реализация Входа и Выхода на сайт компании “ЭкоТакси”

Регистрация

Логин

Пароль

ЗАРЕГИСТРИРОВАТЬСЯ

☰ Эко Такси

Добавьте информацию о себе

Фамилия

Возраст

Email

Телефон

Адрес

ДОБАВИТЬ

Рисунок 20, 21 - Реализация Регистрации пользователей и добавления личной информации.

ЗАКЛЮЧЕНИЕ

Цель курсовой работы заключалась в реализации web-сервиса ЭкоТакси, которое могла бы использовать схожая компания для сбора отчетности, приема заказов и отслеживания работы сотрудников.

Для решения данной задачи был выбран следующий стек технологий: Django Rest Framework, Vue.js, PostgreSQL, что позволило достаточно быстро и удобно реализовать задуманный проект. В результате выполнения курсовой удалось выполнить все вышеставленные задачи, которые соответствуют запросам технического задания. Цель курсовой работы достигнута в полном размере.

В качестве дальнейшего развития проекта можно усовершенствовать дизайн вэб сайта, к примеру, добавить изображения к листу водителей, расширить лист запрашиваемой подробной информации о клиенте. С точки зрения функционала можно добавить регистрацию через социальные сети, а также оплату на сайте, если этот проект “выйдет в свет” и будет использован.

СПИСОК ЛИТЕРАТУРЫ

1. Django Rest Framework. Документация Django Rest Framework [Электронный ресурс]. URL: <https://www.django-rest-framework.org> (дата обращения: 29.06.2020).
2. WebDevBlog. Создание Django API используя Django Rest Framework [Электронный ресурс]. URL: <https://webdevblog.ru/sozдание-django-api-ispolzuya-django-rest-framework-apiview/> (дата обращения: 29.06.2020).
3. Evantotuts+. JWT Аутентификация в Django [Электронный ресурс] URL: <https://code.tutsplus.com/ru/tutorials/how-to-authenticate-with-jwt-in-django--cms-30460> (дата обращения: 29.06.2020).
4. Vue.js. Документация Vue.js [Электронный ресурс]. URL: <https://vuejs.org> (дата обращения: 29.06.2020).
5. Vuetify. Документация Vuetify [Электронный ресурс]. URL: <https://vuetifyjs.com/ru/> (дата обращения: 29.06.2020).
6. Stack OverFlow. Set initial vuetify vue-select value [Электронный ресурс]. URL: <https://stackoverflow.com/questions/51392719/set-initial-vuetify-v-select-value> (дата обращения: 29.06.2020).
7. Asyncee. Продвинутые запросы в Django: сортировка по дате [Электронный ресурс]. URL: <https://asyncee.github.io/posts/advanced-django-querying-sorting-events-by-date/> (дата обращения: 29.06.2020).