**СОДЕРЖАНИЕ**

*ВВЕДЕНИЕ.............................................................................................................4*

*Цели работы..........................................................................................4*

*Задачи.....................................................................................................4*

*Актуальность работы.........................................................................5*

*Анализ существующих решений..........................................................5*

*1. Сбор данных с исходной выгрузки сайтов......................................................5*

*2. Загрузка данных с сайтов................................................................................8*

*3. Получение контактных данных с сайтов. Создание общей базы данных по сайтам и их контактной информации...............................................................11*

*3.1 Получение основной контактной информации.................................11*

*3.2 Получение юридической контактной информации..........................13*

*3.3 Создание общей базы данных по сайтам и их контактам.................................................................................................14*

*4 Анализ общей базы данных по сайтам и их контактной информации.........................................................................................................15*

*5 Создание базы данных связей между компаниями......................................18*

*6 Создание базы данных рейтингов сайтов компаний-автодилеров.......................................................................................................19*

*7 Создание веб-сервиса для оценки безопасности сайтов автодилеров.......................................................................................................21*

*ЗАКЛЮЧЕНИЕ..................................................................................................23*

*СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..........................................25*

*ПРИЛОЖЕНИЯ................................................................................................26*

**ВВЕДЕНИЕ**

В наше время в интернете распространено мошенничество.

Есть множество сайтов, рекламирующих отдельные автосалоны. Если внимательно проанализировать содержимое этих ресурсов, можно понять, что сайты создаются буквально на каждый автосалон, а бывает и на каждую марку авто. Однако, количество настоящих автодилеров сильно ограничено.

Мошенники создают поддельные сайты и закупают большое количество рекламы. Из-за этого ссылки на их ресурсы появляются первыми в ответ на поисковый запрос. Думая, что эти ссылки являются самыми популярными, а значит и самыми безопасными, пользователь переходит на поддельный ресурс и попадается на удочку мошенников, которые завышают цены, оформляют невыгодные кредиты и обманывают покупателя всеми возможными способами.

Сайты-мошенники часто связаны между собой, ведь обычно они создаются в большом количестве, чтобы хотя бы один сайт привлек клиентов. Поэтому поставленную проблему можно решить с помощью рейтинга безопасности сайтов автобизнеса на основе анализа их связей.

**Цели работы**

* Проанализировать базу данных связей сайтов автодилеров.
* На основе анализа связей создать базу данных рейтинга сайтов.
* Создать собственный сервис для проверки безопасности сайтов.

**Задачи**

* Изучить многопоточный метод программирования для оптимизации скачивания данных с интернет-ресурсов.
* Собрать данные с исходной выгрузки сайтов.
* Очистить данные и создать базу данных, состоящую из рабочих ссылок.
* Загрузить данные со всех сайтов из БД (база данных).
* Проанализировать контактные (телефон, почта, адрес, название), юридические (ИНН, ОГРН) и другие атрибуты, чтобы понять, как связаны между собой эти бизнесы и сколько на самом деле у них ЛПР (лиц, принимающих решения)
* Создать базу данных по сайтам и их атрибутам.
* Создать базу данных по связям между сайтами.
* Проанализировать базу данных связей между сайтами.
* Создать базу данных рейтингов безопасности сайтов.
* Запустить web-service для проверки надежности сайтов.

**Актуальность работы**

Сфера автобизнеса сейчас заполонена недобросовестными автодилерами. В наш век информационных технологий мошенники используют интернет как площадку для своей преступной деятельности. Многие пользователи, плохо разбирающиеся в безопасности в сети, попадаются на удочку обманщиков. Поэтому задача создания сервиса для простого и быстрого определения надежности сайтов автобизнеса по рейтингу, выявления недобросовестных продавцов и пресечения их мошеннической деятельности очень актуальна.

**Анализ существующих решений**

Покупатель имеет возможность проверить историю автомобиля, покупая его на специализированных площадках(avito.ru), но там не каждый человек находить нужную ему машину, поэтому многие пользователи ищут их в интернете. В сети есть сайты, описывающие возможные способы обмана и то, как их избежать, но выполнение всех этих инструкций занимает много времени, и покупатели редко ими пользуются. Поэтому решение поставленной задачи является очень перспективным.

1. **Сбор данных с исходной выгрузки сайтов**

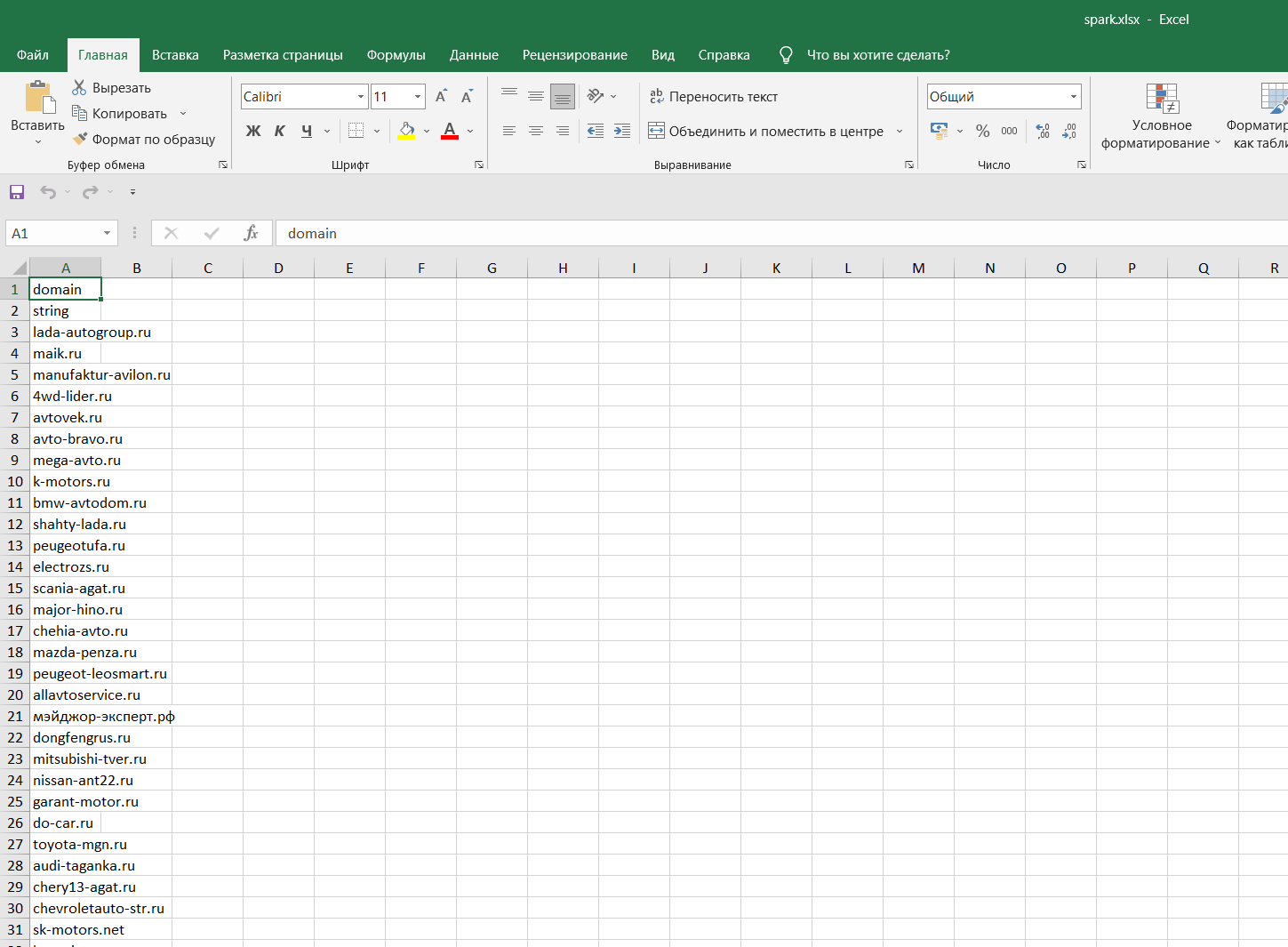
Проект сделан на языке *Python*.

Исходные данные представляют собой выгрузку сайтов, каким-либо образом связанных с автобизнесом. Выгрузка была предоставлена компанией Яндекс. Исходное БД состояло из двух файлов: (рис. 1) и (рис. 2). Две эти базы данных в сумме включают в себя 11623 ссылки.

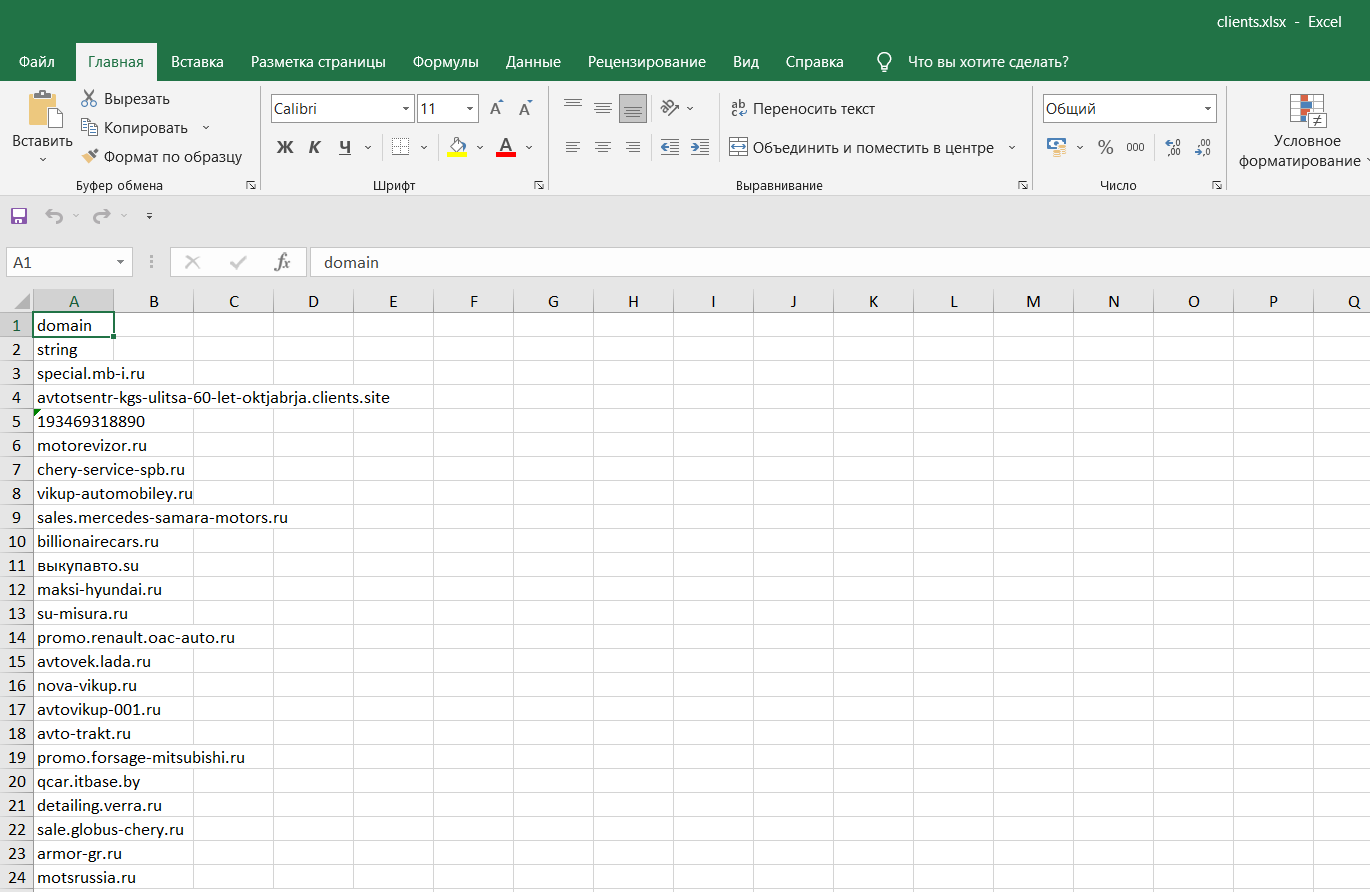
В изначальной выгрузке данные были не отсортированы: встречались нерабочие ссылки, наборы цифр, ссылки на закрывшиеся сайты, ссылки, не имеющие отношения к автобизнесу. Для дальнейшей корректной работы необходимо было очистить данные: убрать наборы цифр, сайты с неправильным доменом и прочие ошибочно попавшие данные. На этом этапе работы еще нет отбора неработающих сайтов, удаляются только те ссылки, которые однозначно не могут быть сайтами.

Для работы с базой данных в формате xlsx был использован модуль csv. Также была использована библиотека pandas для удобства работы с большими данными.

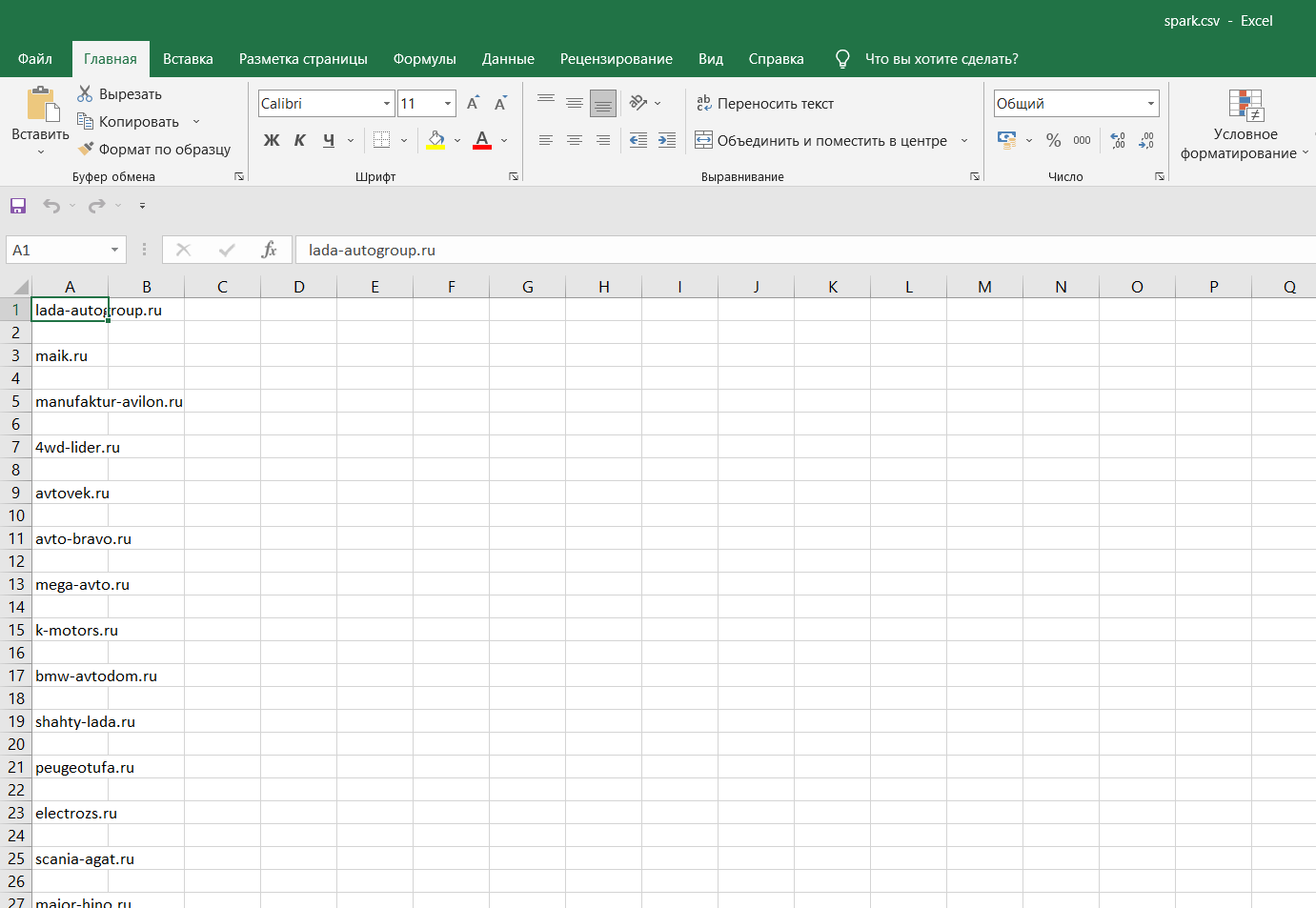
Отфильтрованные данные были записаны в два новых файла: (рис. 3) и (рис.4). Суммарное количество ссылок после первой фильтрации – 8661. Было удалено 2962 некорректные ссылки.



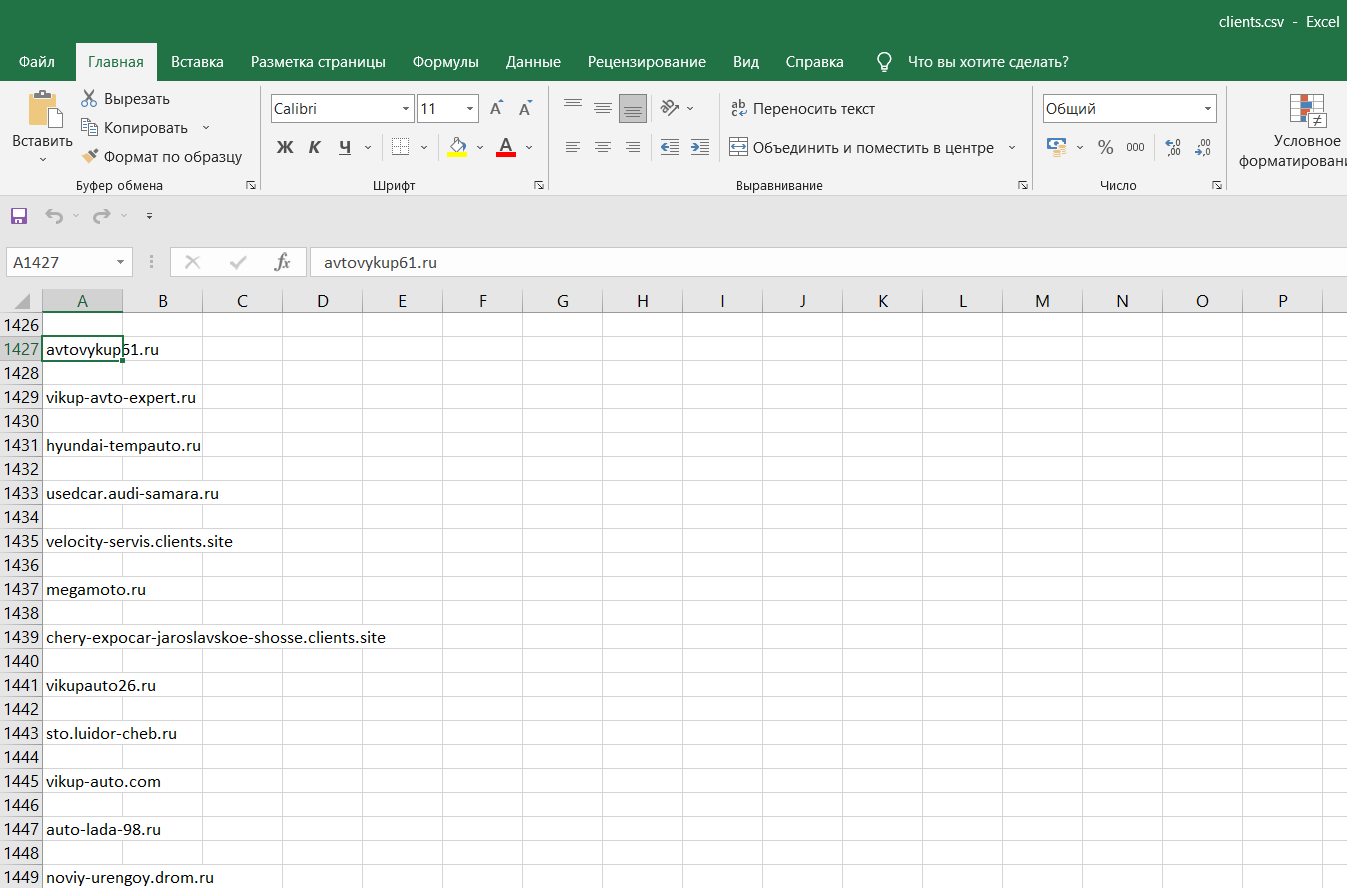
**Рисунок 1. Структура первой части исходной базы данных**



**Рисунок 2. Структура второй части исходной базы данных**

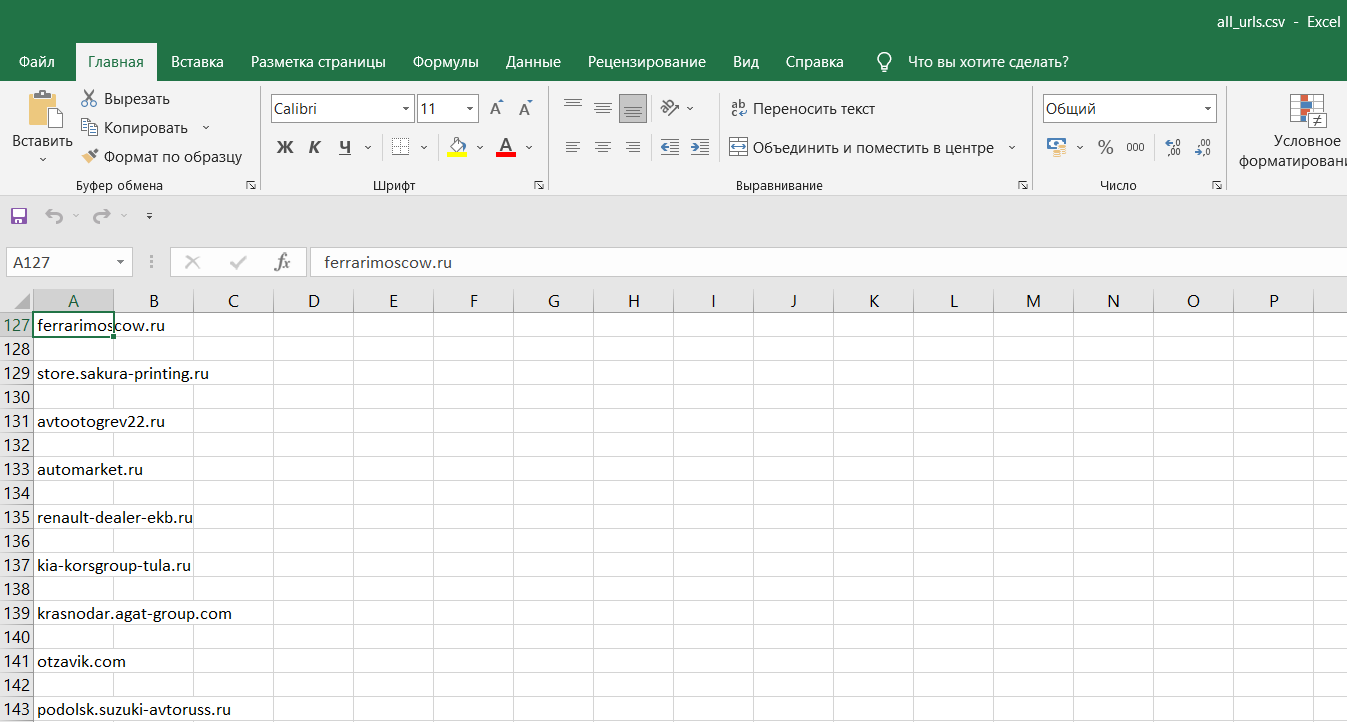


**Рисунок 3. Структура первой части исходной базы данных после фильтрации**



**Рисунок 4. Структура второй части исходной базы данных после фильтрации**

Но работа с двумя базами данных одновременно неудобна, поэтому две базы дынных были объединены в одну общую БД – (рис. 5).



**Рисунок 5. Структура объединенной БД**

Итоги первого этапа работы:

* Отфильтрована исходная выгрузка сайтов: удалены наборы цифр, ссылки с неверными доменами, некорректные ссылки. После фильтрации из 11623 ссылок осталось 8661.
* Собрана единая база данных для всех ссылок.

**2. Загрузка данных с сайтов**

Поставленная задача – составить базу данных со связами сайтов и, проанализировав ее, получить рейтинг безопасности для каждого сайта. Но для нахождения связей между сайтами, нам нужно сначала загрузить контактные (телефон, почта, адрес, название), юридические (ИНН, ОГРН) и другие атрибуты, которые могут быть связаны со сторонними ресурсами в интернете. Для того, чтобы получить эти данные с сайтов, нужно загрузить их в формате html и обработать их, оставив только ту информацию, которая нужна для нахождения нужных атрибутов. Именно загрузить сайты в формате html-страниц было задачей на втором этапе работы.

Использованные инструменты:

* Библиотеки pandas, urllib3, BeautifulSoup, requests, os, Thread.
* ООП (Объектно-ориентированное программирование)

С каждой ссылкой из БД были проделаны следующие операции:

1. Проверка на принадлежность к автобизнесу (вторая фильтрация)
2. Попытка подключения к сайту. Сайты могут быть нерабочие. Для того, чтобы программа не вылетела на подобных ошибках, была использована специальная функция для “ловушки ошибок” (Приложение 1A)*,* а также метод, основанный на стандартном синтаксисе питона: *try – except* (Приложение 1Б)*.*
3. Если подключение успешно, то с сайтом проводятся следующие операции:

3.1) Начальная страница сайта скачивает в HTML-формате с помощью библиотеки BeautifulSoup. Это встроенная в питон библиотека для работы с сайтами и HTML-страницами.

3.2) После скачивания первой(титульной) страницы сайта, с помощью всё той же библиотеки BeautifulSoup, проводится программный анализ ее содержания, для нахождения второстепенных страниц, на которые можно перейти с главной.

*Для чего это нужно:* Если контактные данные (телефоны, почты) часто находятся на главный страницах, то юридические (ИНН, ОГРН) обычно помещаются в “политику конфиденциальности”, или вовсе прячутся очень глубоко (обычно мошенники поступают именно так, ведь им совершенно не нужно, чтобы пользователи могли каким-либо образом проверить их “легальность”). Поэтому, чтобы собрать абсолютно всю контактную и юридическую информацию с сайта, нужно заглянуть на каждую из его страничек.

3.3) Для каждой из вторичных ссылок также производится попытка подключения (как описано в п.2). Если подключение удачно, то производится программный анализ содержания новой странички сайта и ищу ссылки третьего порядка. Таким образом идёт проверка всех страничек второго, третьего и других порядков, пока на них либо вообще не останется ссылок, либо будут ссылки только на те странички, информацию на которых уже была проанализирована (Приложение 1В).

3.4) После программного анализа содержания всех страничек (главной, второго порядка, третьего порядка и так далее) HTML-код был загружен в файл(Приложение 1Г). Все странички сайта были сохранены в одном файле, а не разделялись на страницы и подстраницы для того, чтобы работа по анализу содержания сайта была удобнее.

3.5) Перед сохранением данных с сайта в файл, создается папка с названием сайта, HTML-код которого будет загружен в нее, и эта папка помещается в общую родительскую папку (Приложение 1Г)*.*

***Проблема, возникшая на втором этапе работы и ее решение***

Количество сайтов для загрузки и программного анализа их содержания очень велико – 8661. А так как загрузка информации происходит не только с первой страницы, но и со всех последующих, то количество данных для программного анализа содержания и загрузки увеличивается в n2 раз и более.

*Возникла проблема:*

Время загрузки данных стало очень большим. Компьютер работал всю ночь, но и это не помогло. За 8 часов загрузилась лишь 1/12 всех сайтов.

*Решение проблемы:*

Для решения данной проблемы был использован метод многопоточности (Приложение 1Д).

*Описание метода многопоточности:*

Для загрузки данных с сайтов нам нужно к ним подключаться. Но часто бывает такое, что один из сайтов может зависнуть и время, которое мы затрачиваем на подключение к нему, будет очень велико. В это время вся остальная программа останавливается и ждёт ответа от сервера того сайта. Именно поэтому код программы изначально работал так долго. Метод многопоточности разделяет все процессы, происходящие в программе, на несколько потоков. То есть, программа выполняется не последовательно, а асинхронно. Это позволяет продолжать работу программы и загружать другие сайты, даже если один из потоков в это время “висит”. Данный метод значительно уменьшает время работы программы.

Благодаря использованию многопоточности вся информация со всех сайтов, представленных в объединенной БД, была загружена. Теперь время работы программы составляет меньше минуты, и скачивание такого большого количества данных не вызывает затруднений.

Итоги второго этапа работы:

* Со всех сайтов, представленных в исходной базе данных, была скачена вся информация (HTML-версии сайтов). Информация подготовлена для дальнейшей работы с ней (удобно разложена по папкам и подпапкам).
* Решена проблема медленной работы программы за счёт использования многопоточности.

1. **Получение контактных данных с сайтов. Создание общей базы данных по сайтам и их контактной информации**

После того, как вся информация с сайтов, представленных в исходной БД, была загружена, была поставлена задача получить всю контактную, юридическую и правовую информацию с загруженных сайтов.

*Для чего это нужно:* Наша главная задача – найти связи между сайтами. Сайты могут связываться по номерам телефонов, почтам, юридическим и правовым данным, адресам и другим контактам. Поэтому, чтобы найти связь между сайтами, нужно для каждого ресурса определить список контактных данных, по которым он может быть связан с другими ресурсами в интернете.

Программный анализ содержания сайтов был разделен на две части:

1. Получение основной контактной информации (телефоны, почты).
2. Получение юридической контактной информации (ИНН, ОГРН, адреса, имена владельцев и так далее)

***3.1 Получение основной контактной информации***

Основная контактная информация на всех сайтах по продажам автомобилей – телефоны и почты. Они встречаются на сайтах чаще любых других видов контактов.

Использованные инструменты:

* Библиотеки requests, re
* API специализированного сайта[1]
* ООП (Объектно-ориентированное программирование)

Над каждым сайтом из папки общей родительской папкибыли проведены следующие операции:

1. С помощью модуля ***re***из загруженного HTML-кода сайта были найдены сочетания цифр, являющиеся телефонными номерами (Приложение 2А).
2. Форматирование телефонных номеров (Приложение 2Б).

После получения набора цифр, возможно являющегося телефонным номером, наша задача – привести его в общий формат (+79999999999).

Номера телефонов на разных сайтах различаются. Например, где-то они начинаются с “8”, где-то с “+7”. Но могут быть и другие отличия. Поэтому удобной работы с ними и возможности полноценной правильной фильтрации нужно все номера привести к общему формату.

1. Фильтрация телефонных номеров.

Фильтрация проходит двумя способами:

3.1) Фильтрация номеров телефонов с помощью API специализированного сайта[1] (Приложение 2В).

Подключив это API, можно передать запрос с номером телефона к этому сайту. А он, в свою очередь, проверяет его по общественным базам данных и возвращает в ответ json файл с информацией о том, существует этот телефонный номер или нет. Если номер не существует, то мы его не используем.

Но у этого метода есть недостаток: частота возможных запросов к данному API не может превышать 50 запросов в час. Поэтому для ускорения работы и избежания ошибок, дополнительно была написана собственная функцию для проверки существования телефонных номеров.

3.2) Фильтрация номеров телефонов с помощью встроенной функции (Приложение 2Г).

На сайтах часто встречаются шаблонные номера. Они нужны для того, чтобы пользователь понимал, в каком формате ему нужно вводить свои данные. Но нам шаблонные номера мешают, ведь по ним будут связываться сайты, совершенно не относящиеся друг другу. Поэтому мы оставляем только те телефонные номера, которые не шаблонные.

4) С помощью модуля ***re*** из загруженного HTML-кода были найдены сочетания символов, являющиеся почтами (Приложение 2Д). Основной критерий поиска: наличие символа @.

5) Фильтрация почт (Приложение 2Е).

На сайтах часто встречаются либо шаблоны почт, либо ненастоящие почты, используемые не как контакты, а в других целях. На такие почты нельзя написать, а значит они не являются контактами для связи. Поэтому эти почты не представляют для нас интерес. Такие email-адреса имеют другие домены, отличающиеся от доменов реальных почтовых сервисов. После анализа шаблонных почт, был собран список ненастоящих почтовых доменов, встречающихся на сайтах: **.svg, .png, .jpg, .webp, .bmp, .gif, .css, example, team@tilda.cc**. После фильтрации почт, были оставлены лишь те, которые могут быть использованы как контакты для связи.

6)Уникализация.

На сайте одни и те же телефонные номера и почты могут встречаться по несколько раз. Это будет засорять нашу базу данных. Поэтому, с помощью метода *list(set(list\_of\_values))*,были оставлены только уникальные телефонные номера и уникальные почтовые адреса для каждого сайта.

Итоги программного анализа содержания основной контактной информации:

* Для каждого сайта получен уникальный набор телефонов и почт, находящихся на этом сайте.
* Все данные приведены к общему формату.

***3.2 Получение юридической контактной информации***

Юридическая контактная информация является одним из лучших способов найти связи между сайтами, так как она хранит в себе сведения о адресах, владельцах, датах регистрации бизнеса и других атрибутах.

Используемые инструменты:

* Библиотеки os, re, json, BeautifulSoup, requests
* API специализированного сайта[2]
* ООП (Объектно-ориентированное программирование)

Для получения и проверки юридической информации было выбрано API сайта специализированного сайта[2]. С помощью этого API можно получать всю юридическую информацию о компаниях, имея только их ИНН или ОГРН. В *Приложении 3А* находится описание всех данных, которые были получены с помощью API специализированного сайта[2]. Также в *Приложении 3А* представлены расшифровки всех, встречающихся в работе юридических аббревиатур.

Для получения полной информации о компании нужно сначала узнать ее ИНН/ОГРН, поэтому первая задача состояла в том, чтобы найти эти юридические данные на сайтах автодилеров. После получения этих данных, с помощью API специализированного сайта[2] будет возможно получить всю информацию об интересующей меня компании.

Над каждым сайтом из общей родительской папки были проведены следующие операции:

1. С помощью библиотеки BeautifulSoup HTML-код сайта был приведён в удобный для программного анализа содержания формат: были удалены специальные символы и HTML-тэги. Из всего содержимого файла с HTML-кодом был оставлен только текст, который пользователь может видеть на страницах сайта, потому что именно в нем содержится вся информация по ИНН и ОГРН (Приложение 3Б).

2) Получение информации о компании с помощью API (Приложение 3В).

Далее идёт работы уже с чистым текстом сайта. С помощью модуля ***re***в тексте были найдены всевозможные наборы по 10 и 12 цифр (ИНН компани1 и ИП состоит из 10 цифр, а ИНН юридического лица состоит из 10 цифр), а также всевозможные наборы по 13 знаков (ОГРН состоит из 13 знаков). Является ли набор чисел действительным юридическим контрольным числом, было проверено с помощью API специализированного сайта[2] (Приложение 3Г). Подключив это API, можно передавать запрос с ИНН или ОГРН организации к этому сайту. А он, в свою очередь, проверяет его по базам данных налоговых ведомств, получая открытую информацию о интересующей меня организации, и возвращает в ответ файл в формате json с полной юридической информацией о компании. Полный список того, какая информация была получена, находится в *Приложении 3А*.

У данного API большое ограничение на количество запросов, поэтому с ним не возникаем таких проблем, как с API для проверки телефонных номеров. Но это API может не найти данные о некоторых мелких ИП и юридических лица. Поэтому, чтобы не терять данные и возможные связи, дополнительно были написаны собственные функции для проверки валидности ИНН и ОГРН организаций (Приложение 3Д). Эти функции проверяют правильность юридических данных по “официальному алгоритму проверки контрольного числа ИНН и ОГРН” (Приложение 3Е).

Итоги программного анализа содержания юридической контактной информации:

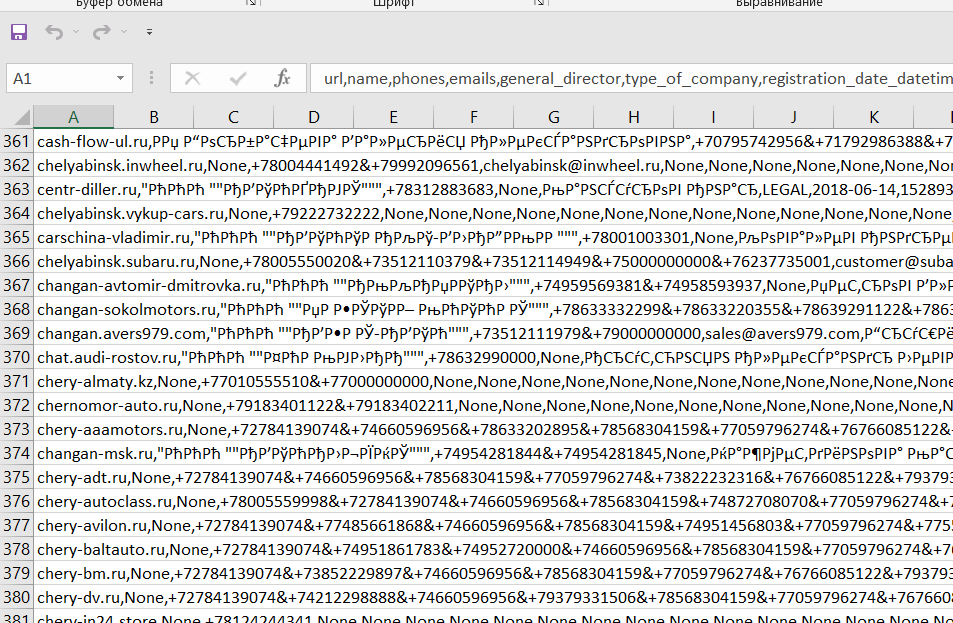
* Для каждого сайта получены ИНН и ОГРН (если таковые на сайте имеются) и по ним получены все юридические данные (Приложение 3А) о компании, которая владеет этим сайтом.

***3.3 Создание общей базы данных по сайтам и их контактам***

После получения основной и юридической контактной информации, с помощью библиотек os и csv, все данные были объединены в новую общую базуданных database.csv(рис.6).

Итоги третьего этапа работы:

* Получена вся информация о телефонах и почтах компаний автобизнеса.
* Получена вся юридическая информация о компаниях автобизнеса.
* Создана общая база данных для всех сайтов, находившихся в исходной объединенной БД. Для каждого сайта записана вся его контактная информация.



**Рисунок 6. Общая база данных по сайтам и их контактам**

Полное описание этой БД содержится в *Приложении 4А.*

1. **Анализ новой общей базы данных по сайтам и их контактной информации**

На прошлом этапе работы были получены все контактные данные по компаниям и сайтам. Но перед тем, как строить взаимосвязи между ними, нужно проанализировать каким образом по каким данным сайты и компании будут связываться между собой, ведь могут быть атрибуты общие для большого количества компаний, например общероссийский классификатор объектов административно-территориального деления (ОКАТО). Одинаковый ОКАТО присваивается всем компаниям одного региона, а значит, в нашей БД они тоже свяжутся. Но эта связь будет неверной, ведь эти компании ни как с друг другом не контактируют. Поэтому был проведен анализ возможных связей между компаниями и оставлены только те атрибуты, которые являются уникальными для каждой компании.

Использованные инструменты:

* Библиотеки network, matplotlib, neo4j
* Сервис для построение графических баз данных Neo4j[3]

Анализировать уникальность связей сайтов в обычной текстовой БД сложно и неудобно. Поэтому для более наглядного представление информации, используемой в анализе, было решено использовать графовую базу данных.

***Графовая база данных*** – это визуализация базы данных, с помощью графа, в котором вершины – это основные элементы БД (в нашем случае сайты), а ребра – элементы, по которым вершины связываются (в нашем случае все контактные данные).

Сначала графовое представление новой общей базы данных database.csv было создано с помощью библиотеки Networkx (Приложение 5А). Но при загрузке большого количество информации, эта библиотека не может обеспечить удобного просмотра графа: все связи перемешиваются и разобраться в информации не представляется возможным. Поэтому для создание графовой базы данных был выбран сервис Neo4j[3].

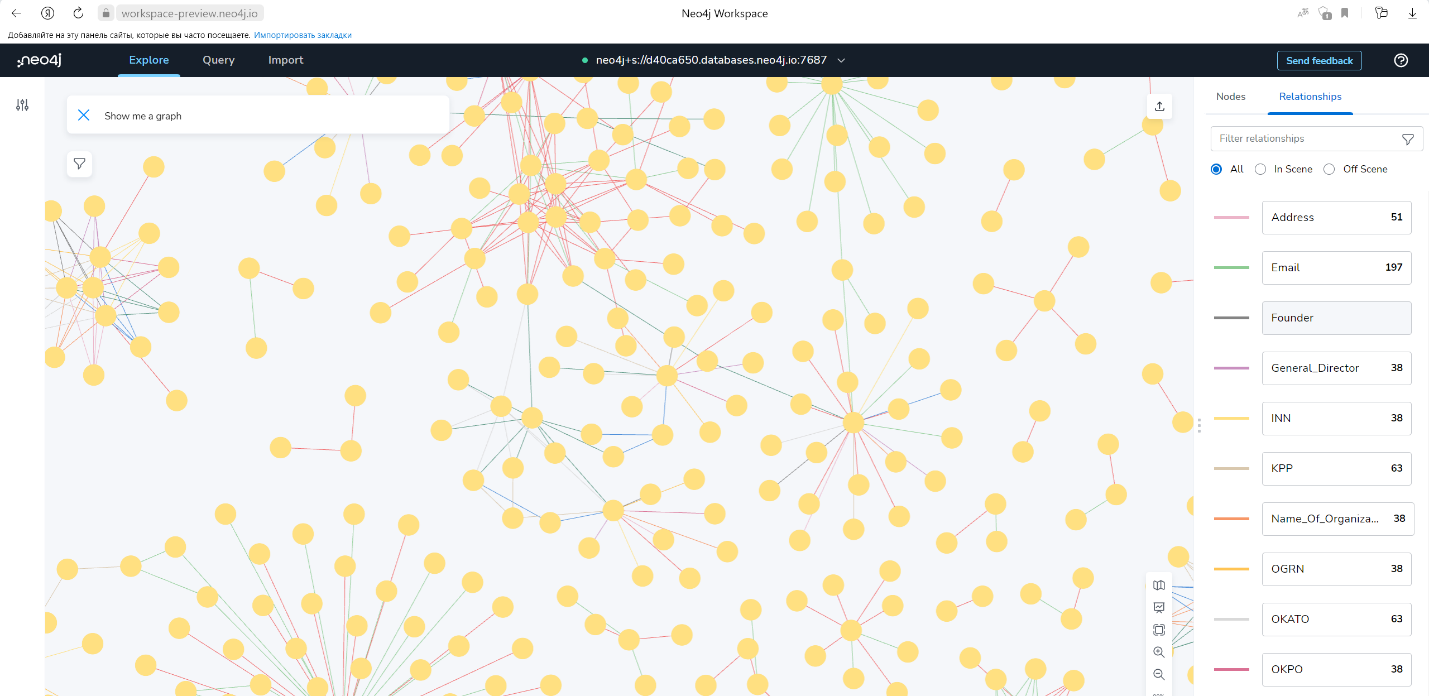
Для создания графовой базы данных на этом сервисе, нужно связать код на питоне с их сайтом. Это было сделано с помощью библиотеки neo4j (Приложение 5Б). Далее были созданы вершины и связи между ними (Приложение 5В).

После создания полной графовой базы данных (рис. 7 и рис. 8), было проведено более детальное изучение связей между сайтами (рис.9) и было обнаружено, что многие из атрибутов, полученные при программном анализе содержания контактной информации сайтов, являются общими для большого количества компаний. Такие атрибуты как ОКАТО, ОКТМО, ОКОГУ связывают автодилеров между собой лишь за счет того, что они находятся в одном регионе, или из-за того, что принадлежат к одному роду деятельности, а значит такие атрибуты нам не подходят, ведь не показывают реальную картину связей между компаниями по уникальным контактным данным.

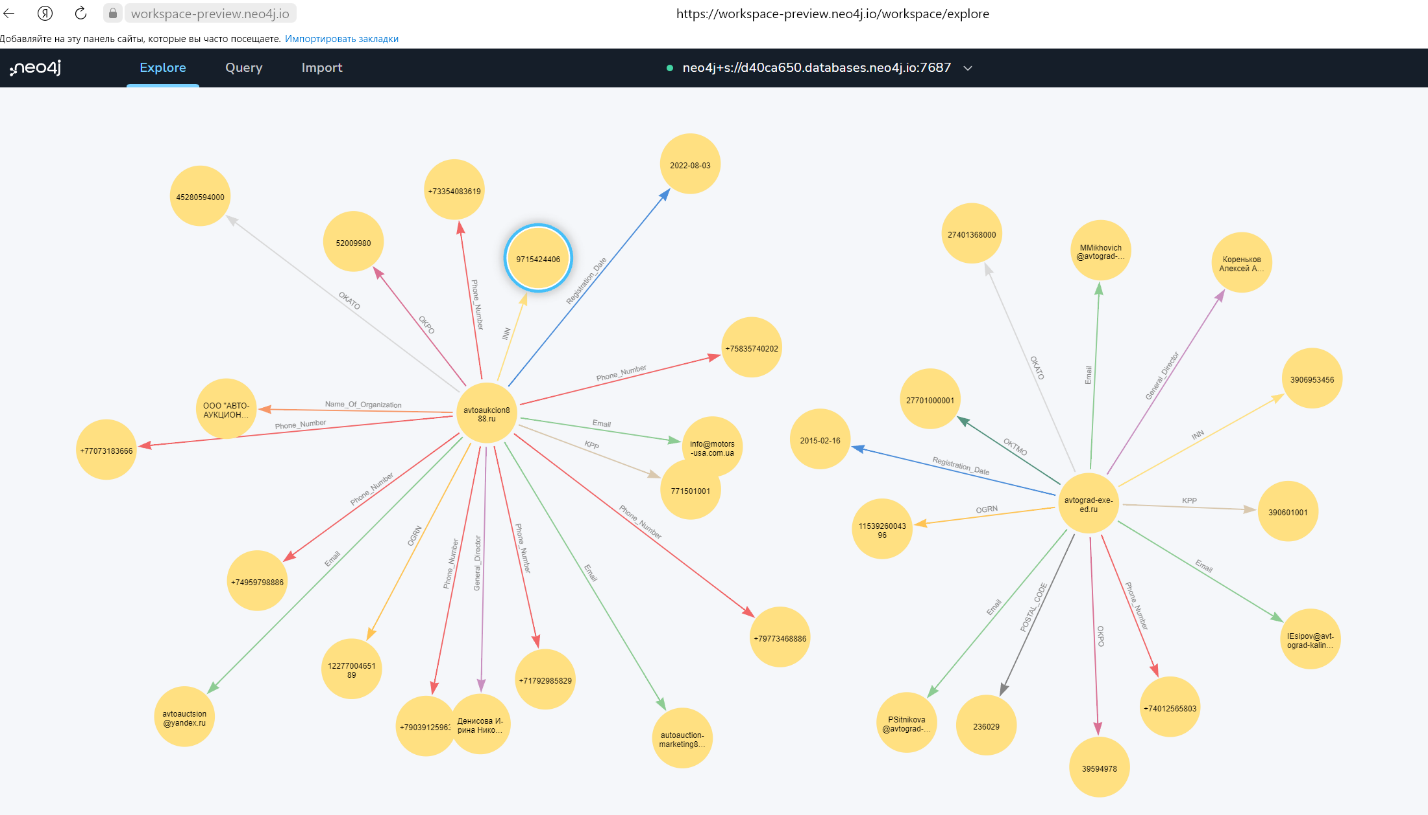
После анализа всех атрибутов с помощью связей в графовой базе данных, были оставлены только уникальные контактные данные: ***название организации, телефонные номера, электронные почты, ФИО генеральных директоров, дата регистрации, ФИО сооснователей, ИНН, ОГРН, КПП, ОКПО, почтовый индекс, адрес компании.***

Итоги четвертого этапа работы:

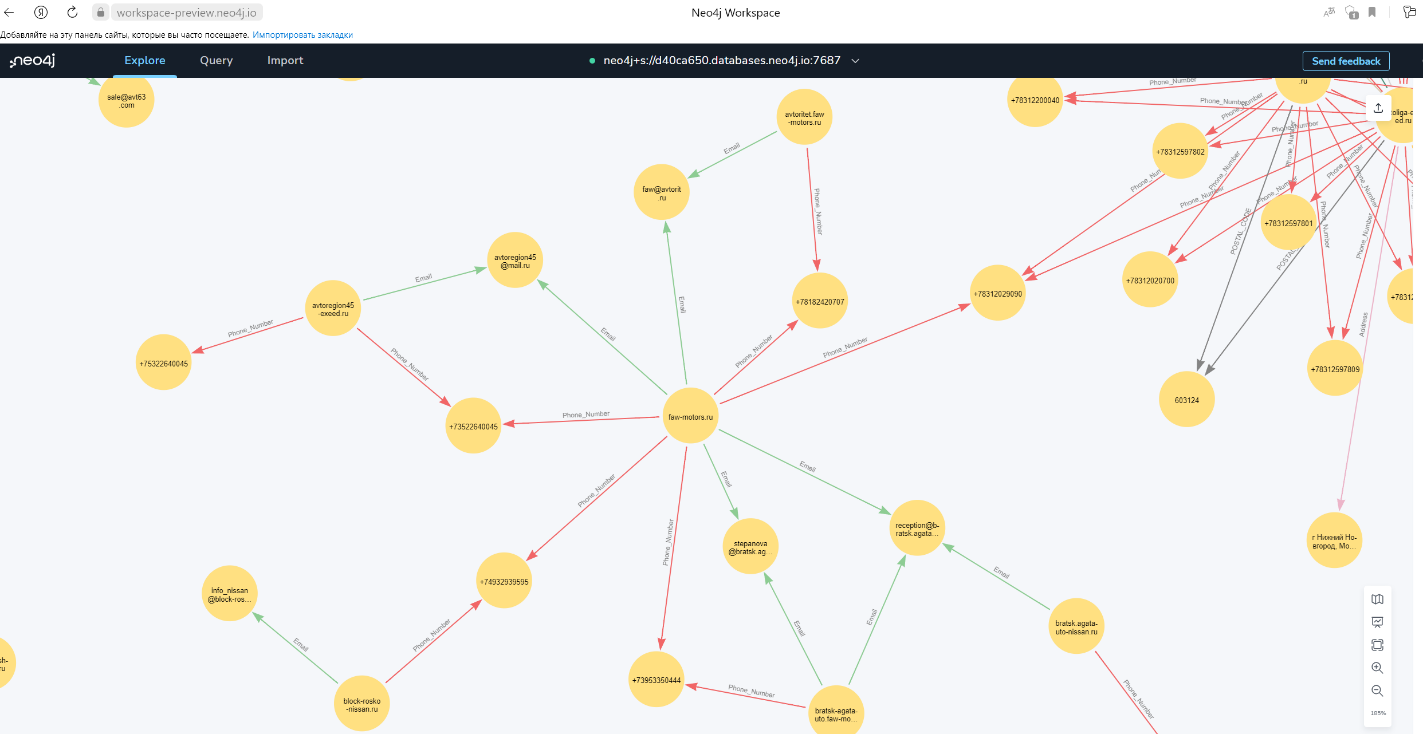
* Создана графовая база данных по связам между сайтами.
* Проанализированы связи между компаниями и оставлены только уникальные контактные данные.



**Рисунок 7. Графовая база данных (общий вид)**



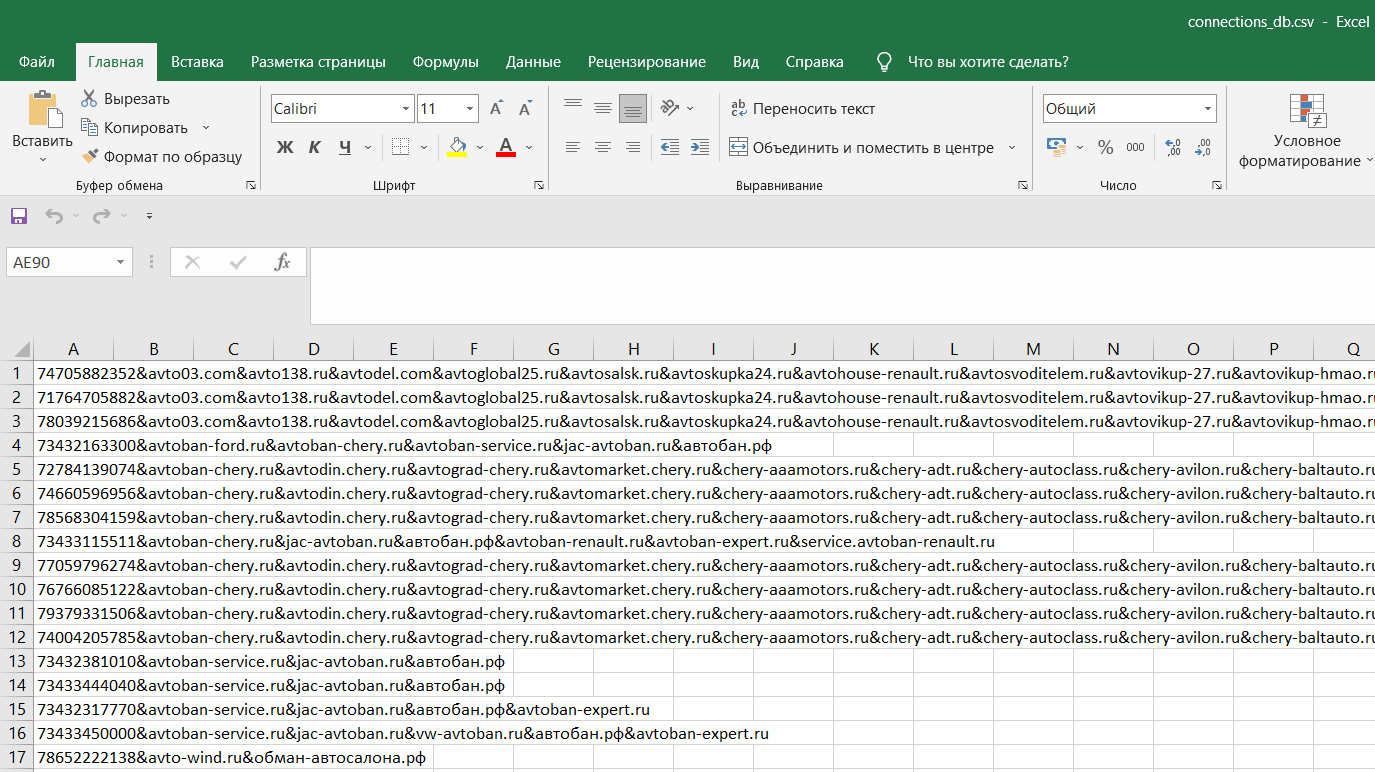
**Рисунок 8. Графовая база данных (пример вершин и ребер)**



**Рисунок 9. Графовая база данных (пример связи между сайтами)**

1. **Создание базы данных связей между компаниями**

На этом этапе работы была создана база данных связей между компаниями (рис.10). Для каждой контактной информации (телефонный номер, почта, ФИО генерального директора и так далее) были обозначены те компании, которые имеют одинаковые данные с этими наименованиями, то есть, если компания “А” и компания “Б” имеют связь по почте “pochta@mail.ru”, то в базу данных записывается следующая информация: ***“pochta@mail.ru” : “A”&“B”***, в качестве разделителя между компаниями(сайтами) используется символ &.



**Рисунок 10. База данных связей между компаниями**

Использованные инструменты:

* Библиотеки pandas, csv

В базу данных была занесена (Приложение 6А) только та информация, которая после анализа на прошлом этапе работы была обозначена как “уникальная контактная информация”.

Итоги пятого этапа работы:

* Создана база данных связей между компаниями.

1. **Создание базы данных рейтингов сайтов компаний-автодилеров**

Для того, чтобы оценивать безопасность компании, нужно понимать ее “рейтинг надежности”.

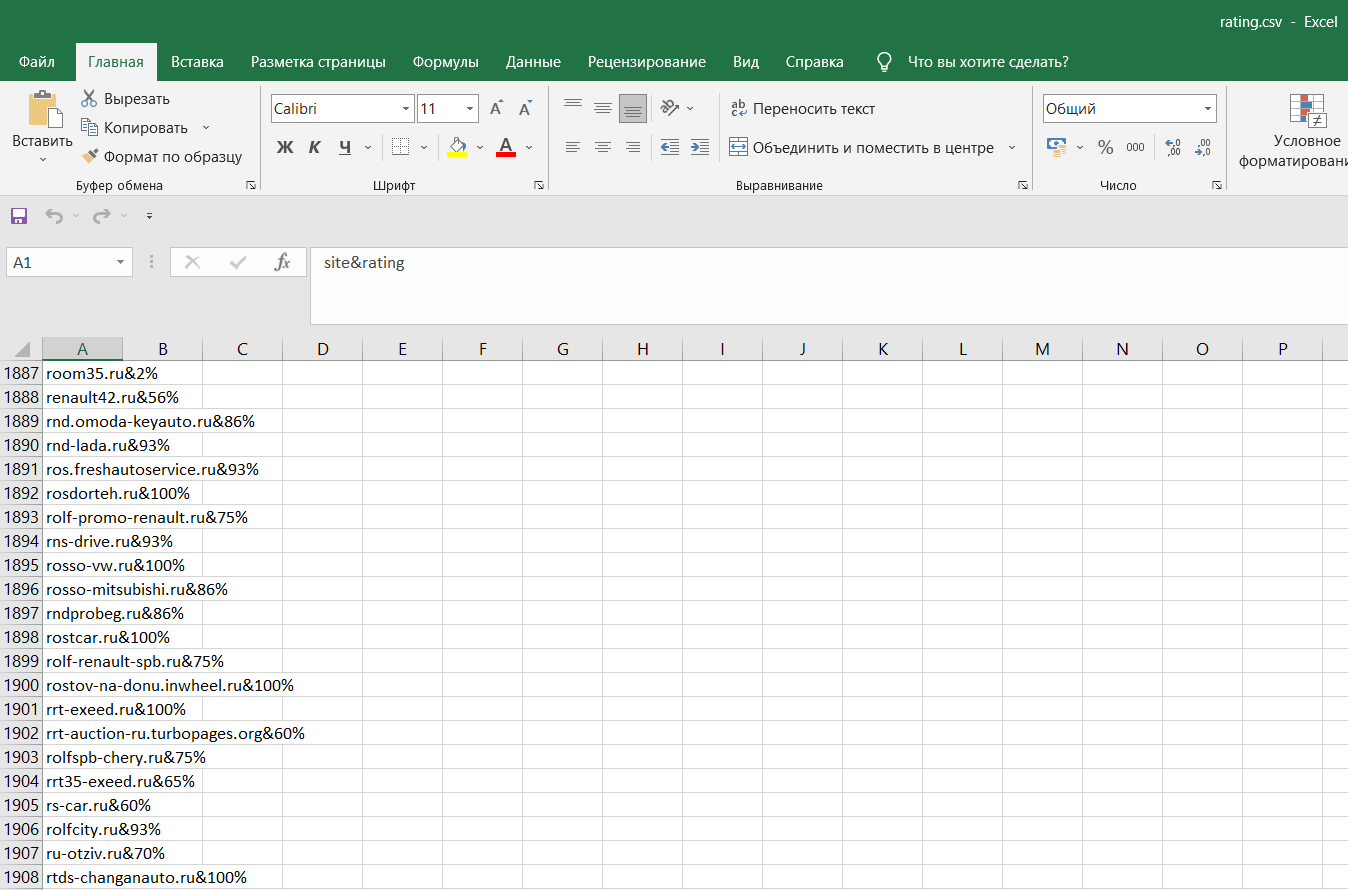
“Рейтинг надежности” – это специальная оценочная система, созданная во время работы над этим проектом, для определения того, насколько сайт безопасен. Этот рейтинг вычисляется по специальной формуле, где каждый сайт изначально имеет “100% надежности”, но этот показатель уменьшается в зависимости от связей данного сайта с “подозрительными ресурсами” (компаниями, занимающимися скупкой подержанных автомобилей, аукционами на покупку зарубежных автомобилей, выкупом битых машин и другими видами неофициальной(серой) деятельности). При нахождении такой связи, рейтинг сайта уменьшается в 0.87. Так проверяются все сервисы, с которыми связана исследуемая компания. Если же на самом исследуемом сайте есть обозначения занятия “подозрительной” деятельностью, то рейтинг сайта уменьшается в 0.6 раз.

Использованные инструменты:

* Библиотеки pandas, os, BeautifulSoup, csv
* “Рейтинг надежности”

Над каждым сайтом из БД связей были проведены следующие операции:

1. Составлен список контактов этого сайта для проверки. В него добавлены данные по 12 атрибутам: ***название организации, телефонные номера, электронные почты, ФИО генеральных директоров, дата регистрации, ФИО сооснователей, ИНН, ОГРН, КПП, ОКПО, почтовый индекс, адрес компании.***
2. Каждый контакт из списка для проверки был исследован по базе данных связей между компаниями (Приложение 7А). Если таковой элемент находится в этой БД, то он добавляется в список связей для проверяемого сайта.
3. Для каждого сайта, который попал в список связей, была проведена проверка их деятельности (Приложение 7Б). С помощью библиотеки BeautifulSoup HTML-код сайта был приведён в удобный для исследования информации формат: были удалены специальные символы и HTML-тэги. Из всего содержимого файла с HTML-кодом был оставлен только текст, который пользователь может видеть на страницах сайта, потому что именно в нем содержится информация о деятельности сайта. Если сайт занимается “подозрительной деятельностью”: скупкой подержанных автомобилей, аукционами на покупку зарубежных автомобилей, выкупом битых машин и другими видами неофициальной(серой) деятельности, то сайт помечается как “подозрительный” и рейтинг проверяемого сайта, который с ним связан, уменьшается в 0.87 (Приложение 7В). Если же на самом исследуемом сайте есть обозначения занятия “подозрительной” деятельностью, то рейтинг сайта уменьшается в 0.6 раз (Приложение 7Г).
4. Проверяемый сайт и значение его рейтинга безопасности были записаны в общую рейтинговую базу данных (рис. 11).



**Рисунок 11. База данных рейтинга безопасности сайтов**

Итоги шестого этапа работы:

* Создан метод вычисления рейтинга безопасности сайтов автодилеров “Рейтинг надежности”
* Создана база данных рейтинга безопасности сайтов автодилеров.

1. **Создание веб-сервиса для оценки безопасности сайтов автодилеров**

На этом этапе работы был создан веб-сервис, расположенный в сети Интернет, с помощью которого любой пользователь может проверить надежность сайта автодилера, на котором он собирается совершать действия, связанные с покупкой или продажей автомобиля.

Использованные инструменты:

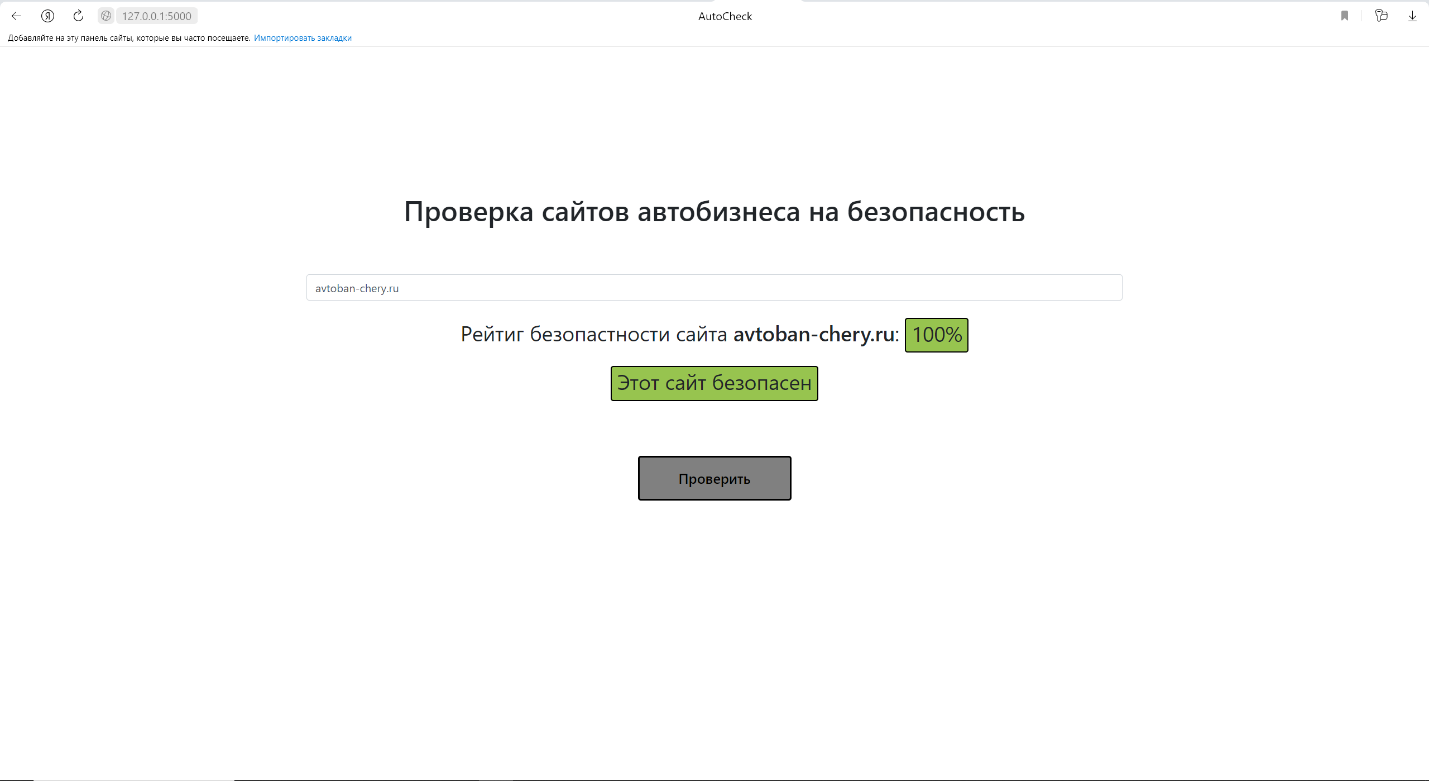
* Библиотеки flask, flask\_wtf, wtforms, pandas
* Инструменты для web-программирования: html, css

Web-сервис был создан с помощью фреймворка Flask (Приложение 8А).

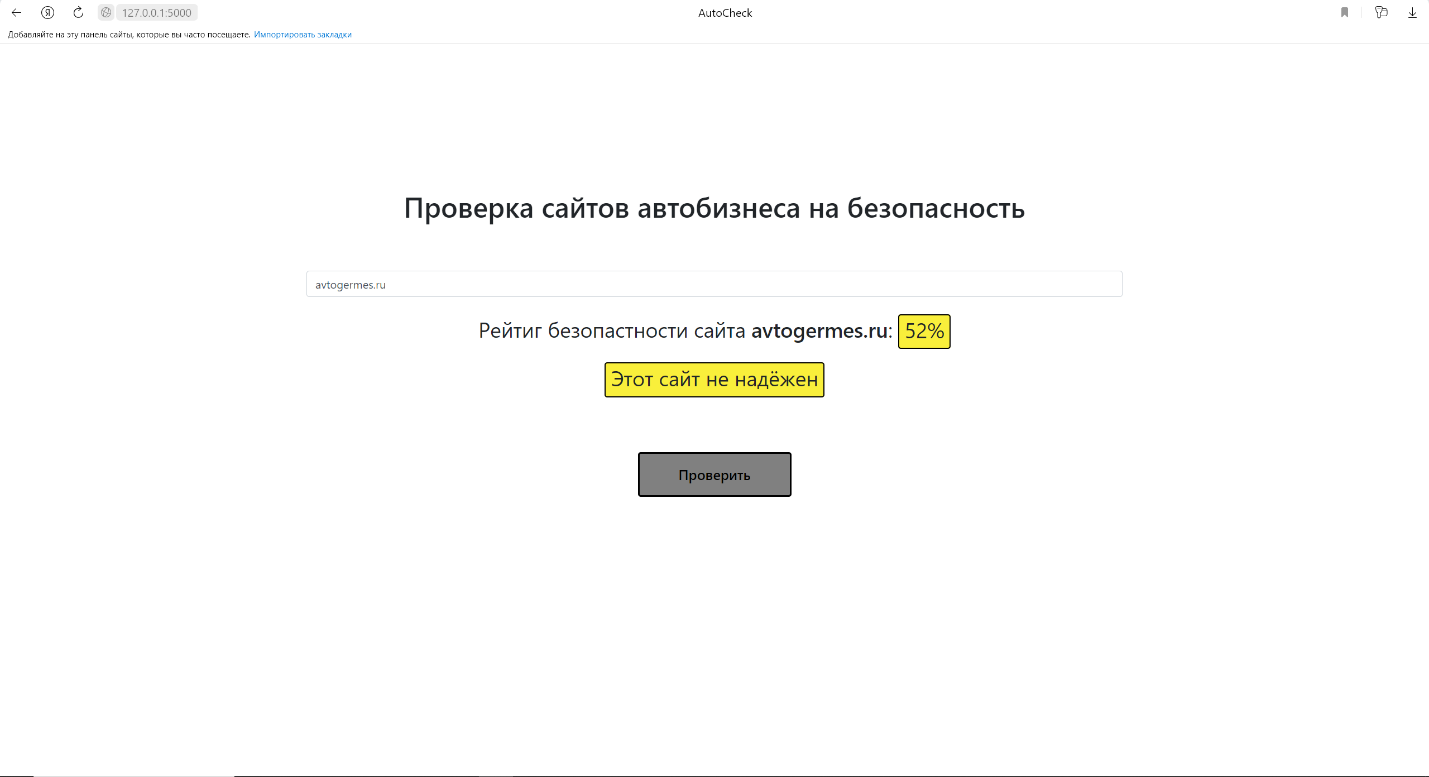
Пользователь вводит сайт автодилера, который он хочет проверить, в специальное поле. Первым делом программа проверяет правильность ввода запроса, если сайт введен с некорректным доменом, то программа сообщает пользователю об этом (Приложение 8Б). Список корректных доменов был вычислен на основе анализа доменов в общей БД всех сайтов database.csv: .com, .site, .рф, .net, .org, .aero, .biz, .coop, .info, .ru. Если сайт введен корректно, то сервис отправляет запрос к базе данных рейтингов сайтов.

Если введенного сайта нет в БД, то программа сообщает, что рейтинг этого сайта не может быть определён (Приложение 8В).

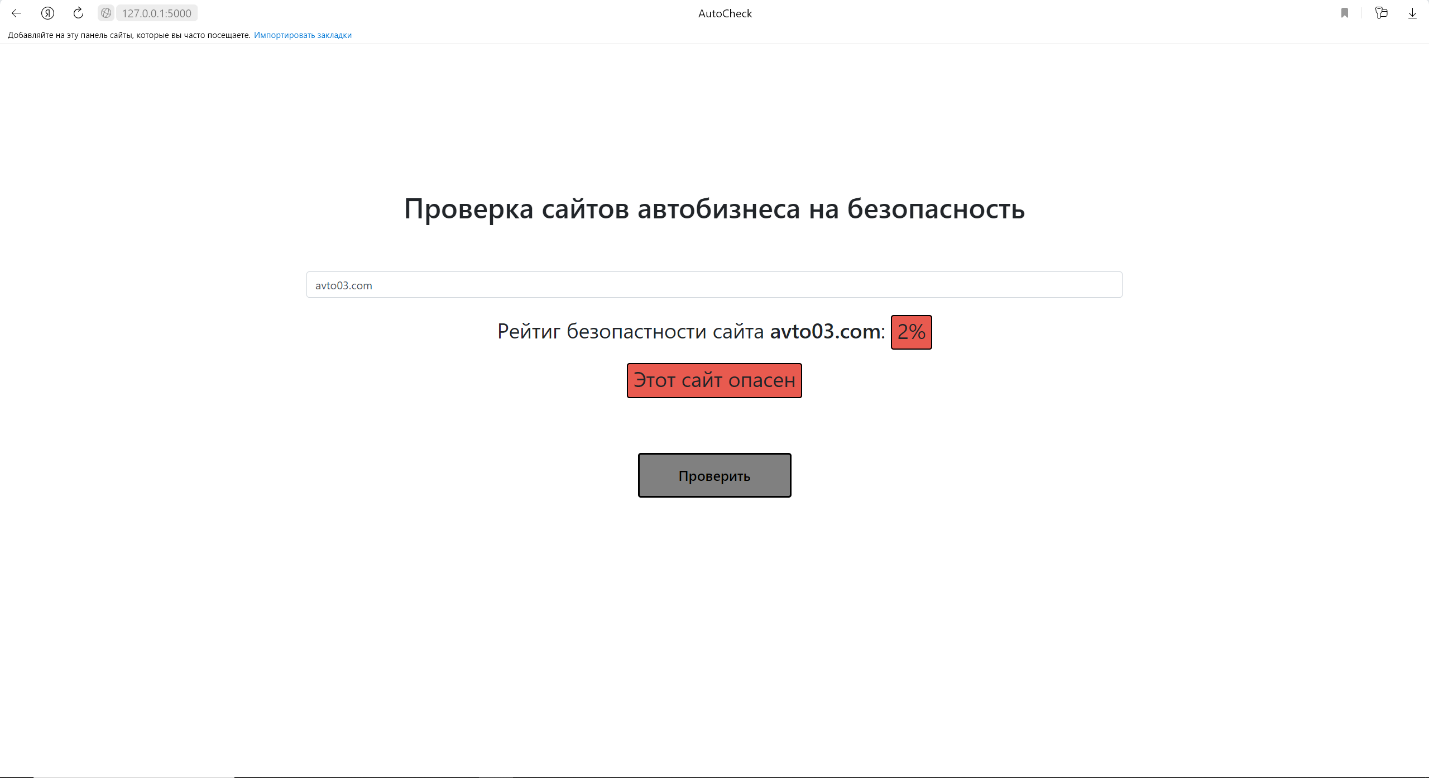
Если же сайт есть в базе данных рейтингов сайтов, то программа передает на сервер рейтинг безопасности сайта, который проверяет пользователь. Сервис выводит название проверяемого ресурса, его рейтинг надежности и анализ, основанный на значении этого рейтинга (Приложение 8Г). Если рейтинг безопасности сайта не менее 80%, то ресурс считается безопасным (рис. 12), если менее 80%, но не менее 35%, то ресурс считается ненадежным (рис .13), если же рейтинг безопасности сайта менее 35 процентов, то ресурс считается опасным (рис.14).



**Рисунок 12. Пример ответа сервиса при проверке сайта с рейтингом не менее 80%.**



**Рисунок 13. Пример ответа сервиса при проверке сайта с рейтингом менее 80%, но не менее 35%.**



**Рисунок 14. Пример ответа сервиса при проверке сайта с рейтингом менее 35%.**

Разметка сайта была написана на HTML (Приложение 8Д). Дизайн сайта и цветное оформление элементов было написано на CSS.

Итоги седьмого этапа работы:

* Создан веб-сервис в сети Интернет, на котором пользователь может проверить безопасность сайта автодилера, на котором он собирается совершать действия, связанные с покупкой или продажей автомобиля.

**ЗАКЛЮЧЕНИЕ**

В ходе проведенной работы достигнуты все поставленные цели, проанализировано более 11500 ссылок на ресурсы, связанные с автобизнесом и создан сервис для проверки безопасности сайтов автодилеров. Найден способ оптимизации скачивания данных за счёт использования многопоточности. Проделана работы по очистке и сортировке данных. Созданы различные базы данных, такие как: объединенная база данных, база данных по связям между компаниями, база данных рейтингов сайтов автодилеров. Построены графовые базы данных и проведен анализ связей между компаниями автобизнеса. Придумана и реализована специальная оценочная система “Рейтинг надежности”.

Созданный сервис, не имеет аналогов на современном рынке, являясь на данный момент единственным способом проверить надежность сайта, на котором пользователь собирается совершать действия, связанные с покупкой или продажей автомобиля.

Созданный сервис ***прост в освоении и использовании***: даже самые неопытные пользователи с легкостью смогут получить на нём всю нужную им информацию.

Вся программная составляющая проекта находится на гитхаб: <https://github.com/Miseadolch/AvtoBusinessesAnalysis/tree/master>

Планирующиеся улучшения сервиса:

- Установка сервиса на полноценный хост в Интернете

- Увеличения количества сайтов в рейтинговой базе данных.

- Улучшение оценочного метода “Рейтинг безопасности”

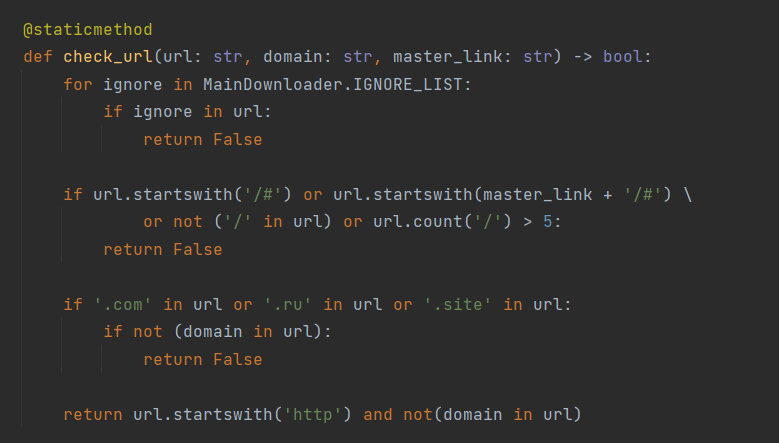
- Применения данного метода оценки надежности производителей в других областях (недвижимость)

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

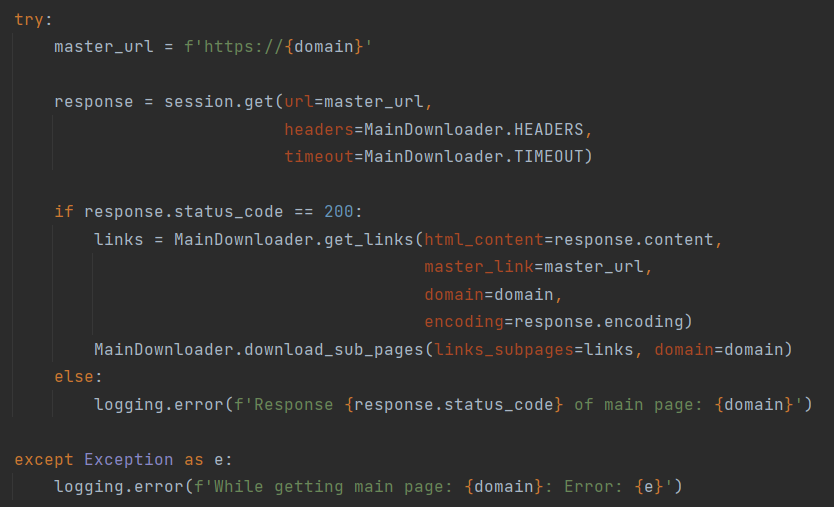
1. API специализированного сайта <https://spys.one/tel/>
2. API специализированного сайта <https://dadata.ru/>
3. Сайт для построение графовых баз данных Neo4j <https://neo4j.com/>

**ПРИЛОЖЕНИЯ**

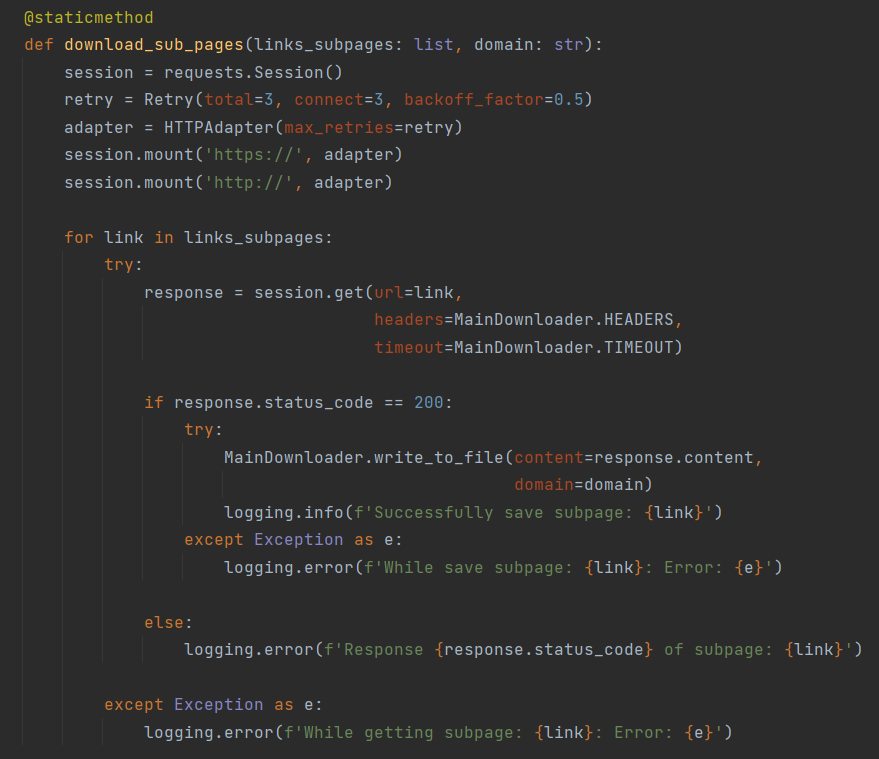
*Приложение 1А*

****

*Приложение 1Б*

**

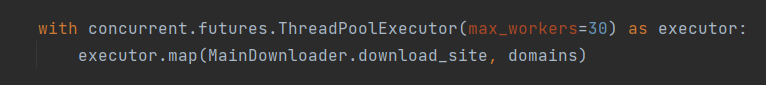
*Приложение 1В*

**

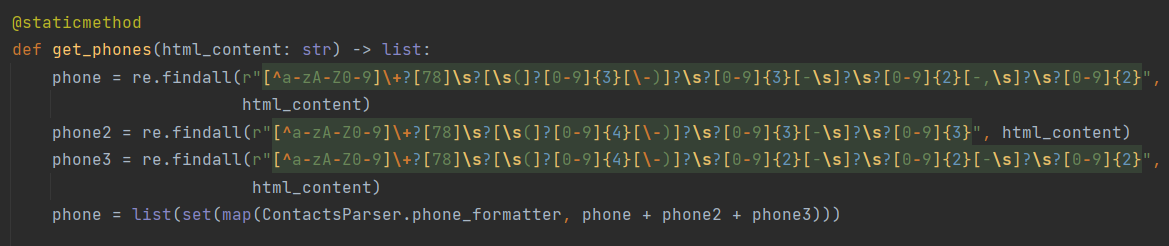
*Приложение 1Г*

**

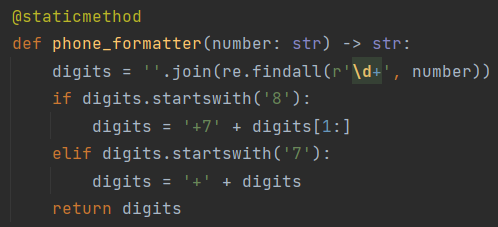
*Приложение 1Д*

**

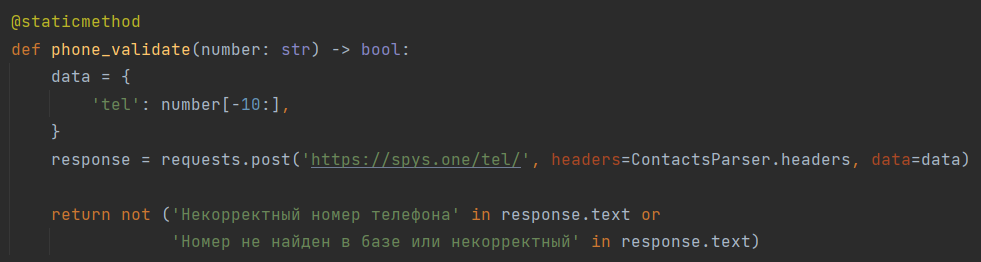
*Приложение 2А*

**

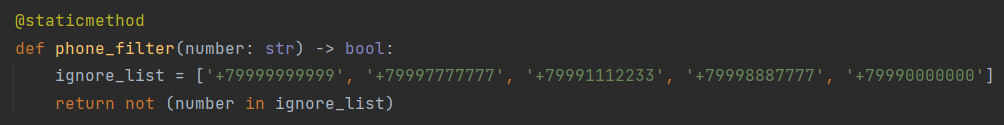
*Приложение 2Б*

**

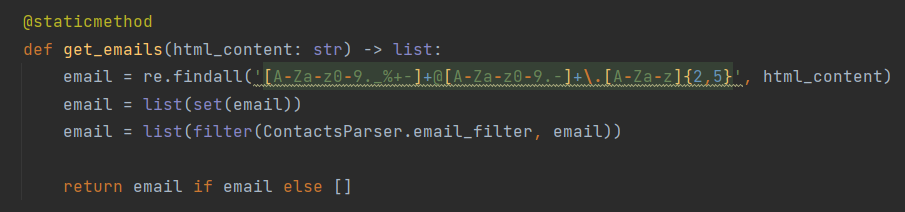
*Приложение 2В*

**

*Приложение 2Г*

**

*Приложение 2Д*

**

*Приложение 2Е*

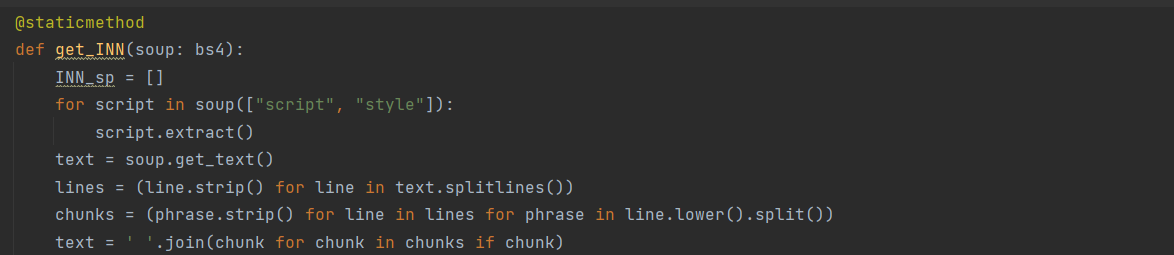
**

*Приложение 3А*

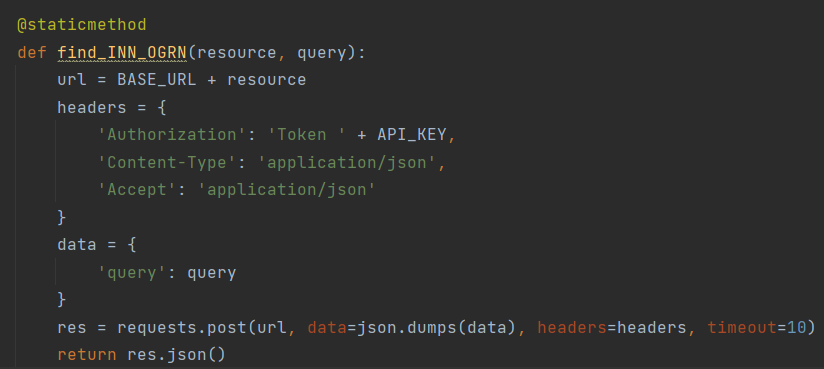
* Название организации
* ФИО генерального директора компании
* Тип компании
* Дата регистрации компании
* Сооснователи компании
* ИНН
* ОГРН
* КПП
* ОКПО
* ОКАТО
* ОКОГУ
* Почтовый индекс местонахождения компании
* Полный адрес местонахождения

*Расшифровка всех юридических аббревиатур находится в приложении 4А*

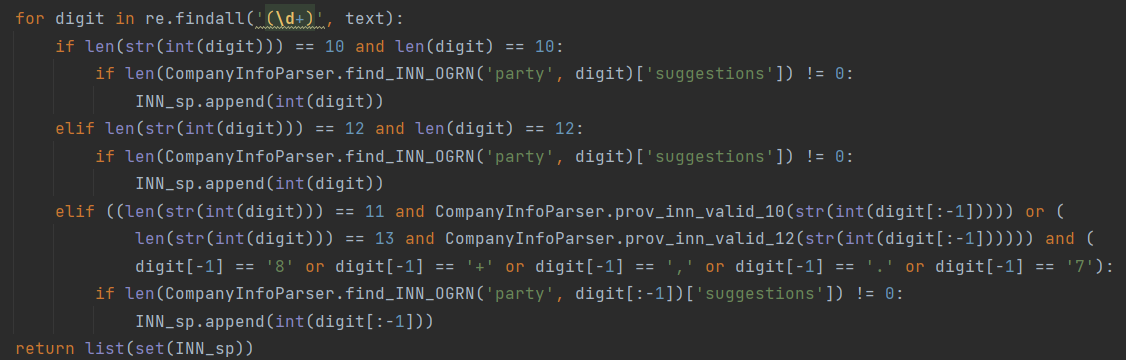
*Приложение 3Б*

**

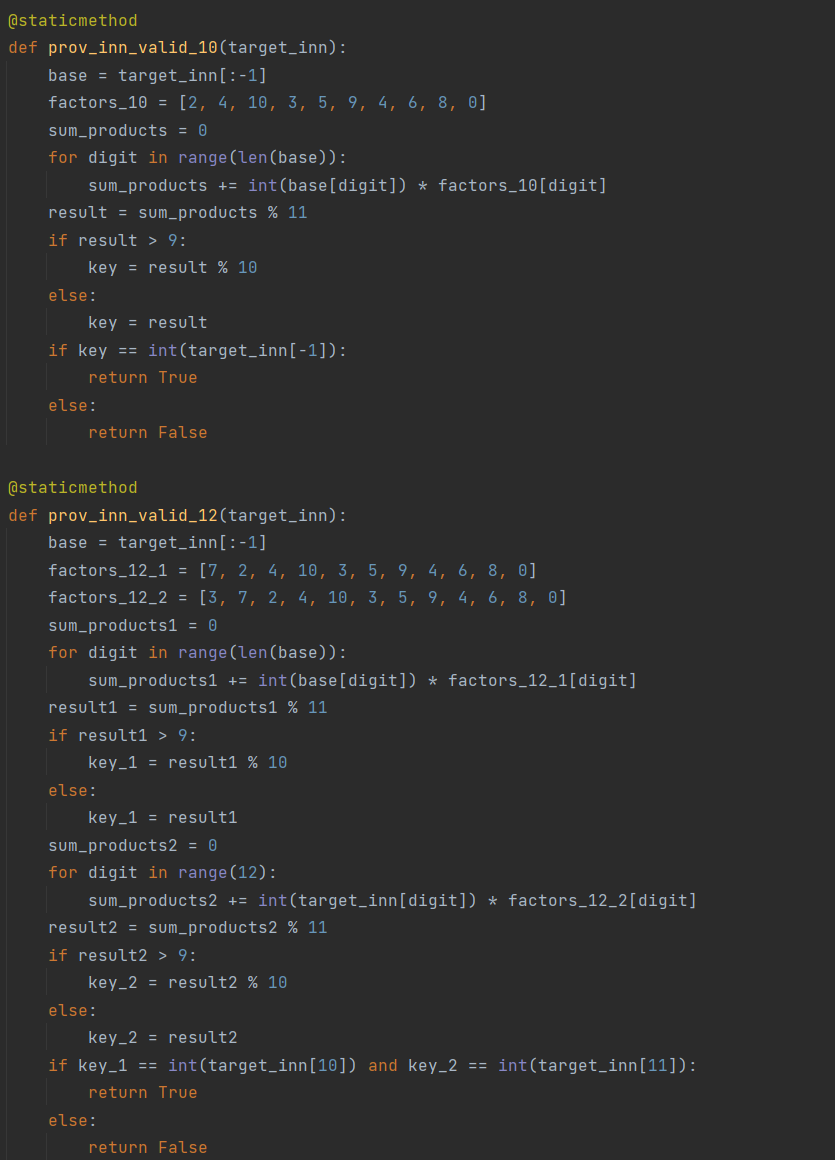
*Приложение 3В*

**

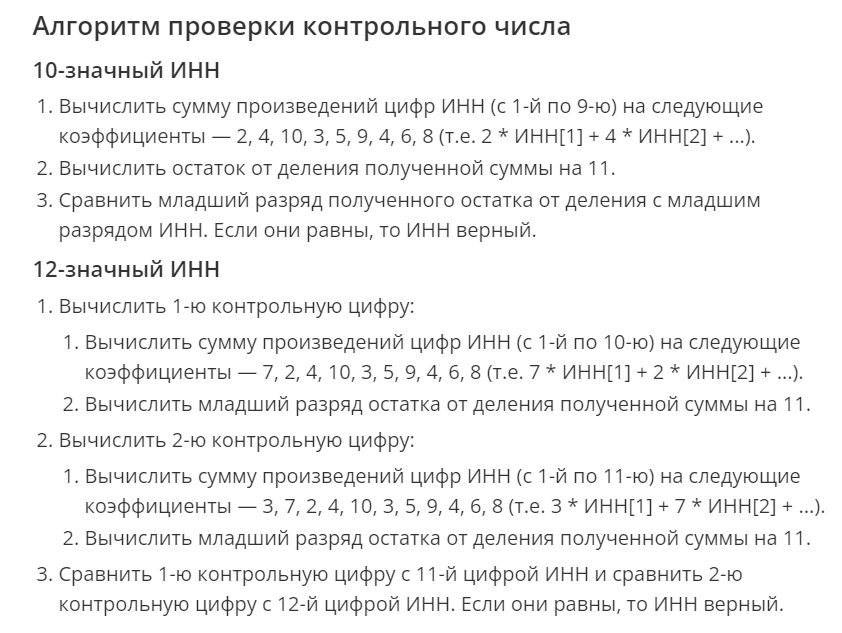
*Приложение 3Г*

**

*Приложение 3Д*

**

*Приложение 3Е*

**

*Приложение 4А*

**## База данных**

[База данных] (/database.csv) располагается в корне репозитория в [формате csv] (https://ru.wikipedia.org/wiki/CSV)

**## Поля**

```url``` - Адрес сайта комапнии *(домен + доменная зона)\_*

```name``` - Официальное название компании *(например, ООО "АВТОЗАПЧАСТИ", если несколько, то через разделитель &)\_*

```phones``` - Контактные номера телефонов *(если несколько, то через разделитель &)\_* в одном формате +7XXXXXXXXXX

```emails``` - Адреса электронных почт *(если несколько, то через разделитель &)*

```general\_director``` ФИО генерального директора компании *(например, Иванов Иван Иванович)\_*

```type\_of\_company``` Тип компании *\_(LEGAL/INDIVIDUAL/None)* LEGAL - юридическое лицо, INDIVIDUAL - индивидуальный предприниматель (ИП)

```registration\_date\_datetime``` - Дата регистрации компании в формате YYYY-MM-DD

```registration\_date\_timestamp``` - Дата регистрации компании в формате timestamp *(секунды от UNIX)\_*

```founders``` - Сооснователи компании *(если несколько, то через разделитель &)*

```inn``` - ИНН: идентификационный номер налогоплательщика *(если несколько, то через разделитель &)\_*

```ogrn``` - ОГРН: основной государственный регистрационный номер *(если несколько, то через разделитель &)\_*

```kpp``` - КПП: код причины постановки на учет *(если несколько, то через разделитель &)*

```okpo``` - ОКПО: общероссийский классификатор предприятий и организаций *(если несколько, то через разделитель &)*

```okato``` - ОКАТО: общероссийский классификатор объектов административно-территориального деления *(если несколько, то через разделитель &)*

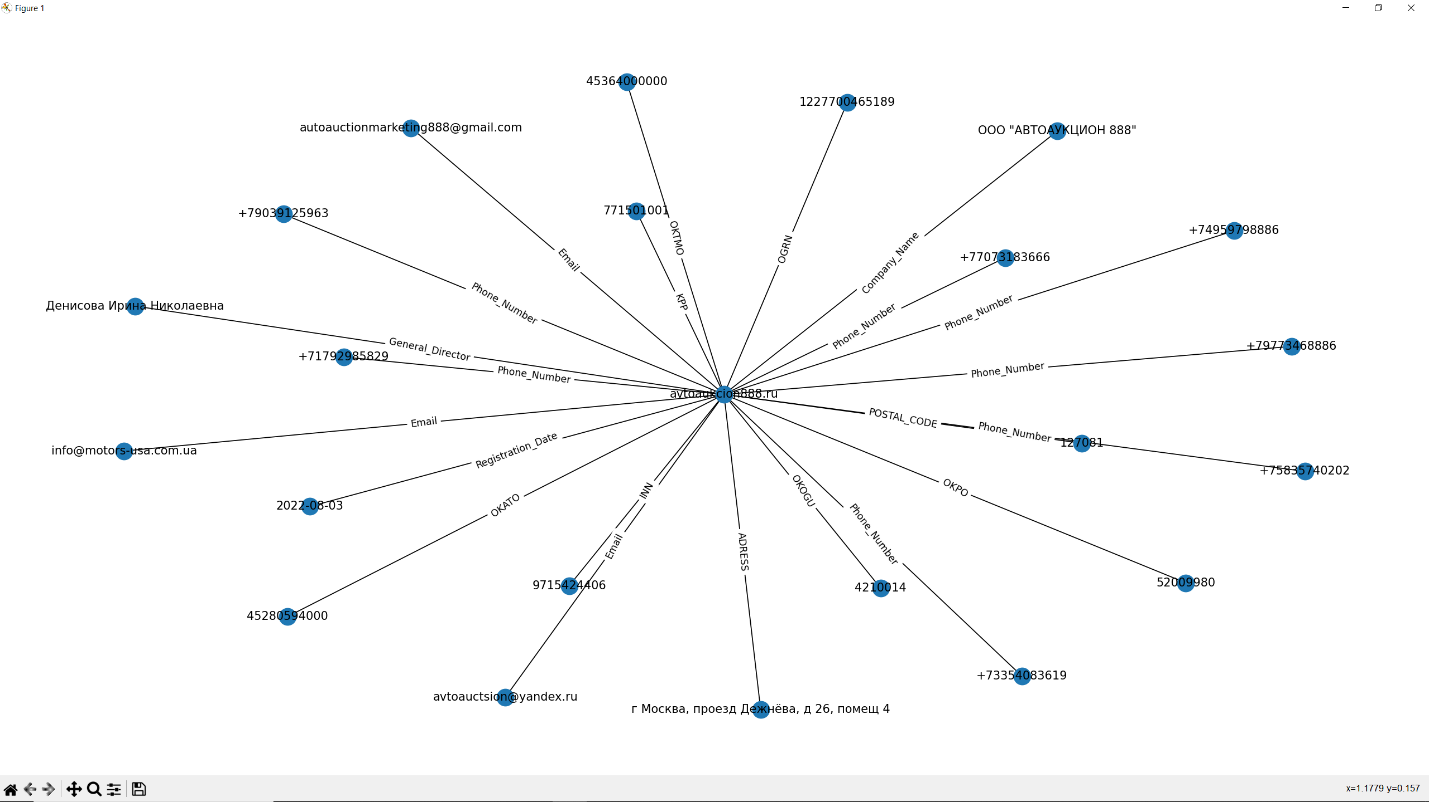
```oktmo``` - ОКТМО: общероссийский классификатор территорий муниципальных образований *(если несколько, то через разделитель &)\_*

```okogu``` - ОКОГУ: общероссийский классификатор органов государственной власти и управления *(если несколько, то через разделитель &)*

```postal\_code``` - Почтовый индекс местонахождения компании *(если несколько, то через разделитель &)\_*

```address``` - Полный адрес местонахождения компании *(например, г Москва, ул Обручева, д 30/1 стр 2, если несколько, то через разделитель &)\_*

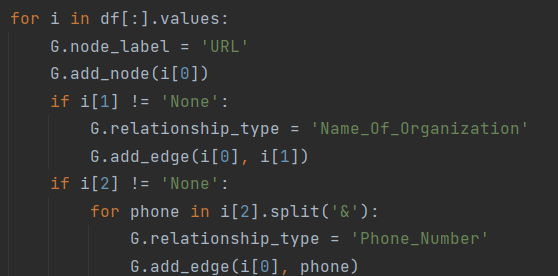
*Приложение 5А*

**

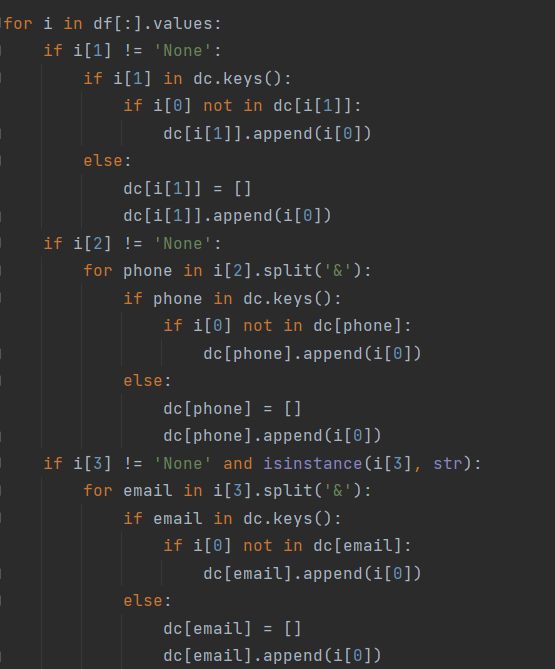
*Приложение 5Б*

**

*Приложение 5В*

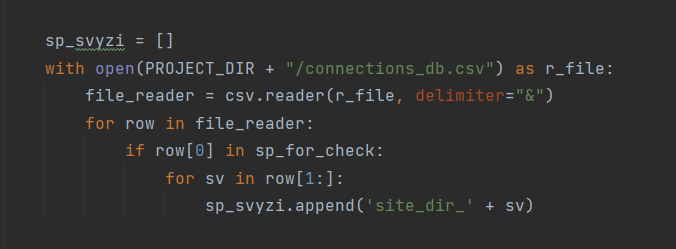
**

*Приложение 6А*

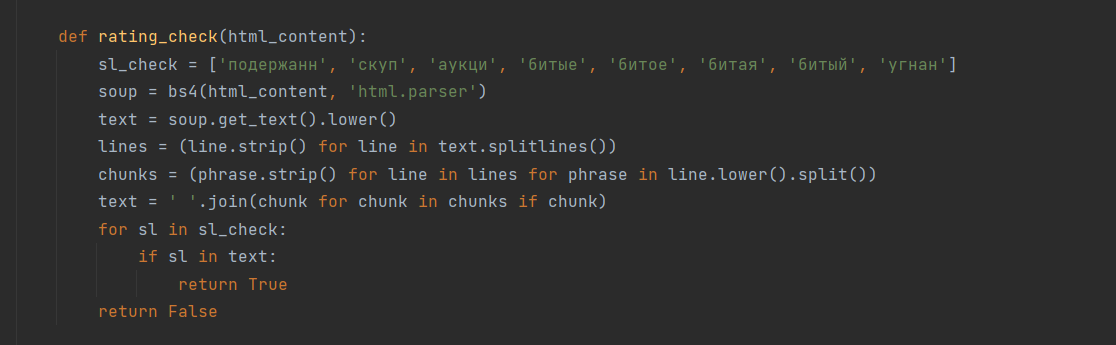
**

**

*Приложение 7А*

**

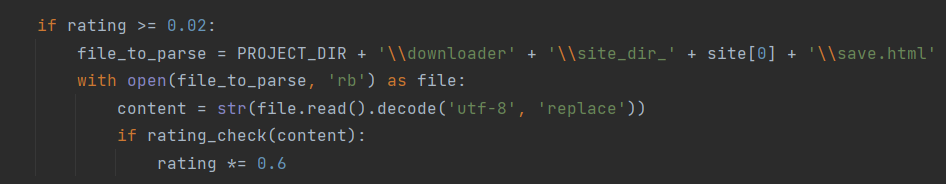
*Приложение 7Б*

**

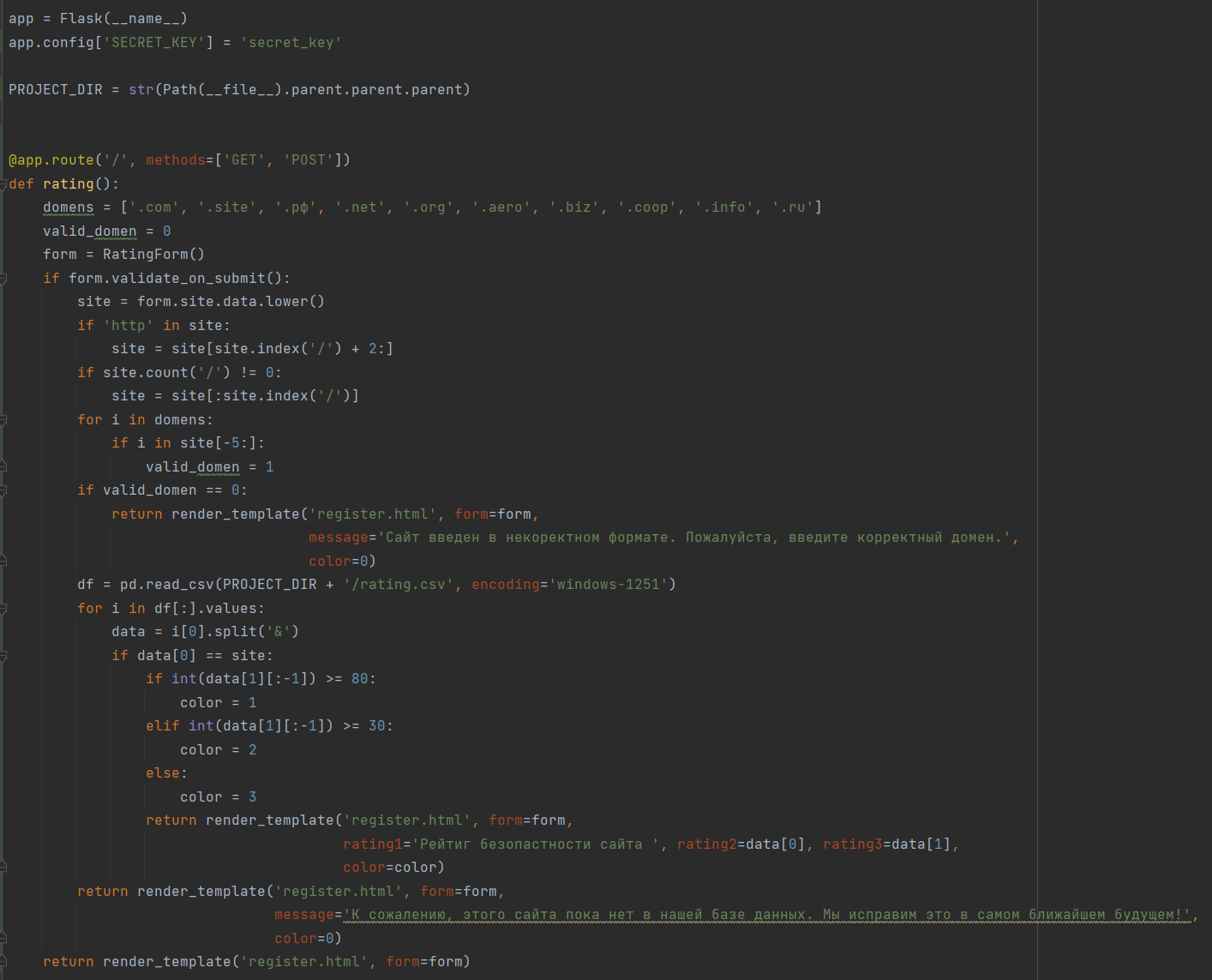
*Приложение 7В*

**

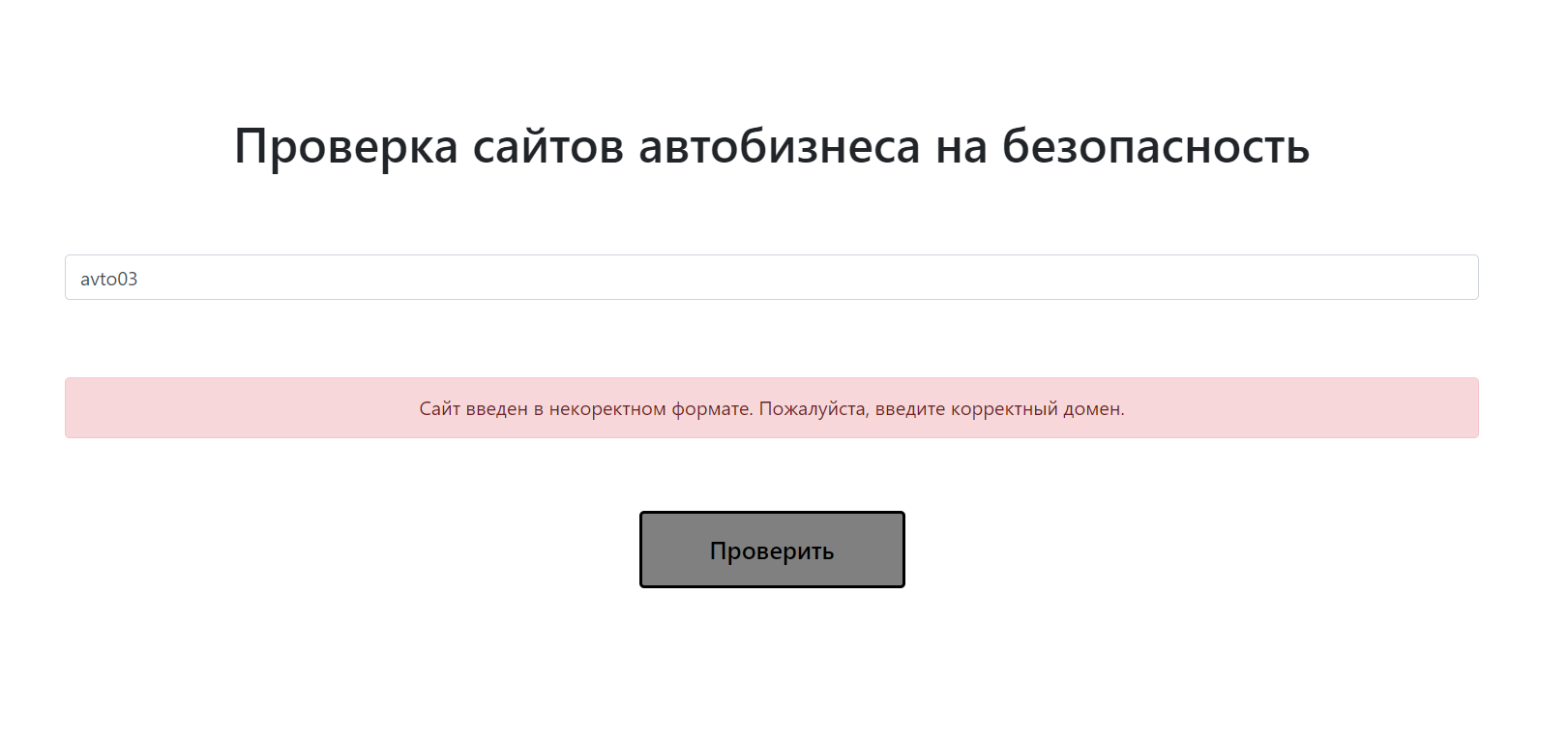
*Приложение 7Г*

**

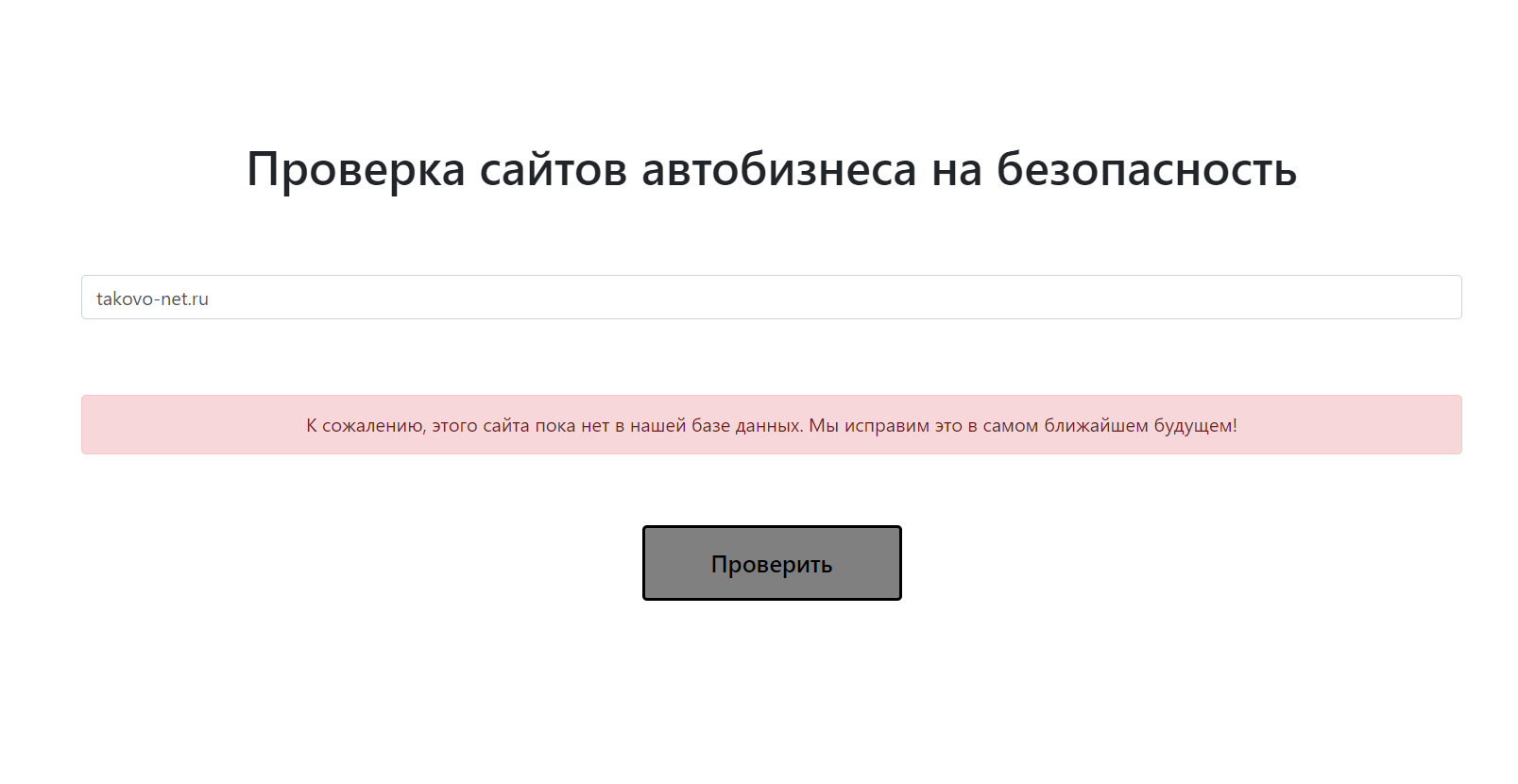
*Приложение 8А*

**

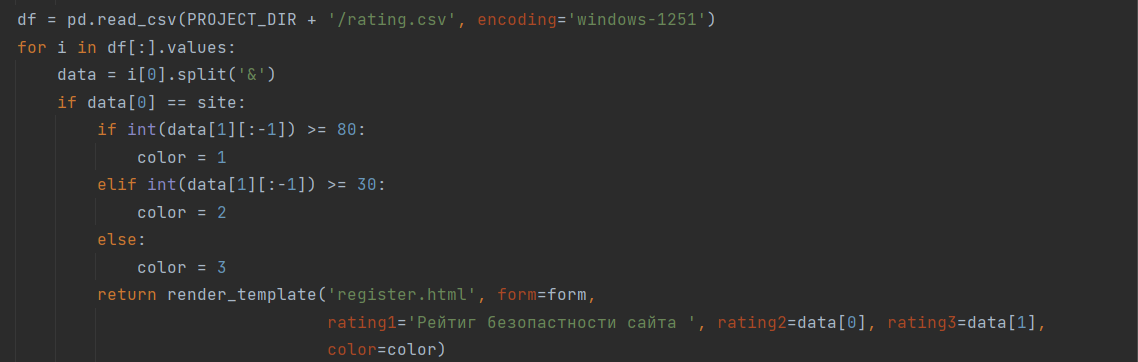
*Приложение 8Б*

**

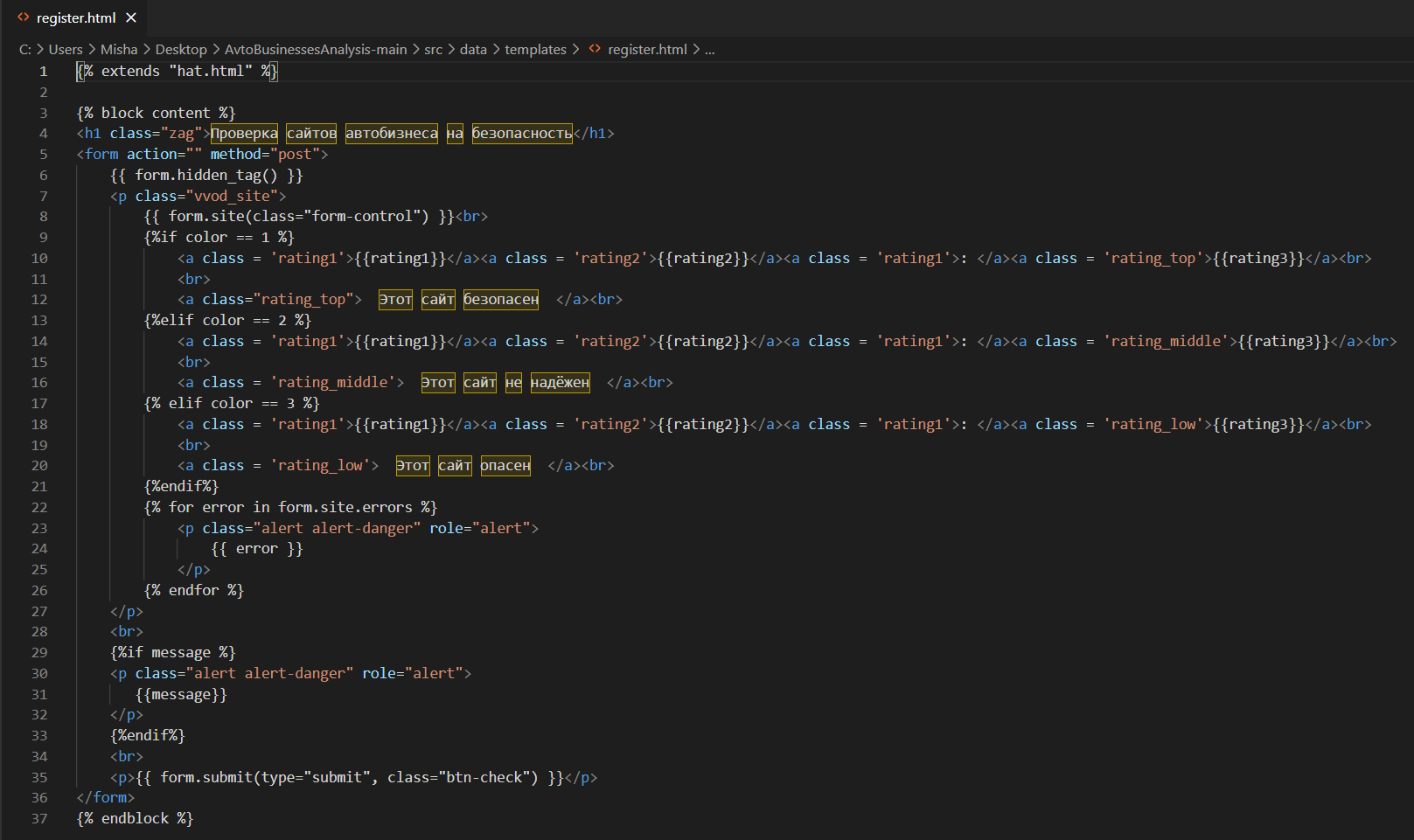
*Приложение 8В*

**

*Приложение 8Г*

**

*Приложение 8Д*

**