**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«ТЮМЕНСКИЙ индустриальный университет»**

Высшая школа цифровых технологий

Кафедра Математики и прикладных информационных технологий

**КУРСОВОЙ ПРОЕКТ**

**Дисциплина:** «Интеллектуальные системы и их проектирование»

**Тема:** «Разработка парсера для агентства недвижимости»

**Выполнил:**

Студент 4 курса группы РИСб-20-1

направления 45.03.04 «Интеллектуальные

системы в гуманитарной сфере»

Халикова С.В.

**Руководитель:**

доцент, канд. пед. наук

Спирин И.С.

Тюмень, 2023 г.

**Аннотация**

Курсовой проект содержит 33 страницы, 10 источников, 11 рисунков.

Ключевые слова: Python, парсер, агентство недвижимости, PyCharm.

Целью данного курсового проекта является – разработка парсера для агентства недвижимости.

Объектом проекта является – парсер для агентства недвижимости.

Предметом проекта является – процесс разработки парсера для агентства недвижимости.

Информационная база исследования: статьи, учебные пособия и научные работы в области создания парсеров, работы в Python.

Результат: разработан парсер для агентства недвижимости.

Курсовая работа выполнена в текстовом редакторе Microsoft Office Word 2016.

**Техническое задание**

* разработать концепцию парсера для агентства недвижимости;
* разработать дизайн парсера для агентства недвижимости;
* разработать функциональную часть и интерфейсную часть парсера для агентства недвижимости;
* произвести тестирование на работоспособность парсера.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc165579301)

[1 Анализ и описание предметной области 6](#_Toc165579302)

[1.1 Общая информация о парсерах 6](#_Toc165579303)

[1.2 Обзор конкурентов 8](#_Toc165579304)

[1.2.1 Обзор парсера «ЦМНС» 8](#_Toc165579305)

[1.2.2 Обзор парсера «INPAS» 10](#_Toc165579306)

[1.3 Перспективы создания парсера для агентства недвижимости 11](#_Toc165579307)

[2 Проектирование парсера для агентства недвижимости 14](#_Toc165579308)

[2.1 Концепция парсера для агентства недвижимости 14](#_Toc165579309)

[2.2 Концепция интерфейса 15](#_Toc165579310)

[2.3 Структура базы данных 19](#_Toc165579311)

[3 Разработка парсера для агентства недвижимости 21](#_Toc165579312)

[3.1 Выбор языка программирования и инструментов 21](#_Toc165579313)

[3.2 Написание функционального кода страниц 25](#_Toc165579314)

[ЗАКЛЮЧЕНИЕ 32](#_Toc165579315)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33](#_Toc165579316)

**ВВЕДЕНИЕ**

С развитием интернет-технологий и цифровизации процессов в сфере недвижимости стало необходимым для агентств недвижимости иметь эффективные инструменты для автоматизации сбора данных о доступных объектах недвижимости. В настоящее время множество агентств используют агрегаторы недвижимости – специализированные веб-сервисы, которые собирают информацию о недвижимости со множества различных источников, облегчая поиск и сравнение объектов для потенциальных клиентов.

Однако, для агентств недвижимости возникают сложности при извлечении и анализе данных с таких агрегаторов. Структура веб-страниц может быть различной, что затрудняет процесс автоматического сбора и обработки информации. В связи с этим возникает потребность в разработке специализированных инструментов, способных проводить синтаксический анализ данных с сайтов агрегаторов недвижимости и извлекать необходимую информацию.

Цель – разработка парсера для агентства недвижимости.

Объектом работы является парсер для агентства недвижимости.

Предмет – процесс разработки парсера для агентства недвижимости.

Задачи данной работы:

* проанализировать предметную область;
* разработать концепцию парсера для агентства недвижимости;
* разработать дизайн парсера для агентства недвижимости;
* разработать функциональную и интерфейсную часть парсера для агентства недвижимости;
* провести тестирование и оптимизацию разработанного парсера для агентства недвижимости.

1. **Анализ и описание предметной области**
   1. **Общая информация о парсерах**

Парсер – это программное обеспечение, которое преобразует структурированные данные из одного формата в другой. Он используется для извлечения информации из сайтов, файлов, баз данных и других источников данных.

Парсинг предлагает инструменты, которые помогают извлекать нужные вам значения из любых форматов данных. Извлеченные сведения сохраняются в отдельном файле – либо на локальном компьютере, либо в облаке или на хостинге, либо напрямую в базе данных. Это происходит автоматически.

Парсер не запрещенный метод сбора и обработки информации. Часто даже сайты госслужб предоставляют данные для публичного использования. Поскольку сбор такой информации – это обработка огромных массивов данных, на помощь приходят парсеры.

Есть два основных метода парсинга: нисходящий и восходящий. Обычно они различаются последовательностью, в которой создаются узлы синтаксического древа.

«Сверху вниз»: при нисходящем методе парсер выполняет поиск сверху, с начального символа в коде, и ищет подходящие ему синтаксические связи. Таким образом, древо синтаксического анализа разрастается сверху вниз – в направлении более детальной разбивки.

«Снизу вверх»: восходящий парсер начинает снизу, с самого нижнего символа строки, а затем устанавливает всё более крупные синтаксические связи. Это делается, пока не будет достигнут начальный символ кода.

Применение парсера:

Парсинг используется для преобразования текста в новую структуру в следующих случаях.

* чтение программного кода – Java, SQL и других языков программирования. Этим занимаются парсеры баз данных. Парсер предоставляет программе структуру данных, с помощью которой можно сгенерировать машинный код;
* чтение HTML-кода с помощью парсера исходного кода страницы. Для компьютера HTML-код – это просто строки символов, которые должны быть проанализированы в браузере. Парсер предоставляет описание веб-страницы как структуры данных, которые затем компонуются в нужном порядке и выводятся в файл;
* анализ XML-разметки. Специальные парсеры XML отвечают за обработку этого типа документов и готовят содержащуюся в них информацию для дальнейшего использования;
* чтение URL-адресов и интернет-протоколов HTTP и HTTPS. Парсеры разбивают сложные схемы URL-адресов и протоколов на иерархическую структуру;

Преимущества парсера:

* автоматизация процесса обработки и анализа больших объемов данных;
* ускорение работы с информацией, так как парсер может обрабатывать данные быстрее и более эффективно, чем человек;
* улучшение точности обработки данных, так как парсер может выполнять задачи без ошибок и искажений;
* возможность обновления данных автоматически и регулярно;
* уменьшение риска ошибок и оптимизация работы с данными.

Недостатки парсера:

* необходимость постоянного обновления и настройки парсера для корректной работы с различными источниками данных;
* риск блокировки доступа к сайту, если парсер не соблюдает правила использования сайта или его API;
* ограничения на обработку сложных и динамических веб-страниц;
* возможность получения неточных или неполных данных в случае ошибок в работе парсера;
* необходимость в технических знаниях для настройки и использования парсера.
  1. **Обзор конкурентов**
     1. **Обзор парсера «ЦМНС»**

CMNS.ru – это онлайн сервис, на котором есть любой город. На сайте предоставлены все виды недвижимости

CMNS работает онлайн-режиме, доступ к нему имеется круглосуточно.

Информация обновляется достаточно быстро, что позволяет отработать информацию.

Сайт является платным 5000 рублей – месяц, что является основным недостатком.

Конкурентный сайт парсера для агентства недвижимости предлагает широкий выбор объявлений о продаже и аренде недвижимости, актуальные цены и возможность фильтрации по различным критериям. Навигация на сайте удобная и интуитивно понятная, что позволяет быстро находить нужную информацию.

Есть фильтрация данных, предоставлено на Рисунке 1.1 *–* Фильтр ЦМНС.

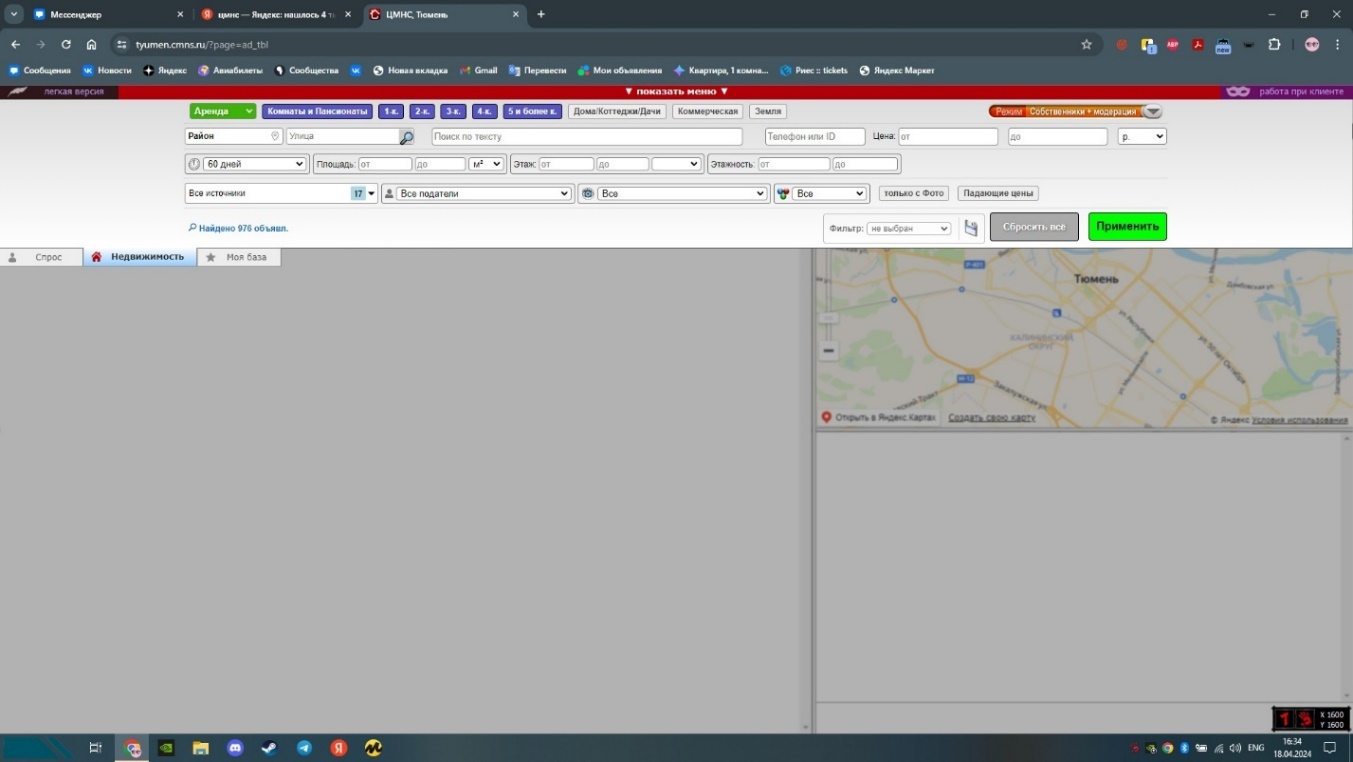


Рисунок.1.1 – Фильтр ЦМНС

Фильтр в парсере – позволяет отбирать определенные элементы или данные из входящего потока информации и обрабатывать их или передавать дальше для дальнейшей обработки. Фильтр работает на основании определенных правил или условий, которые задаются заранее. В случае с ЦМНС – это данные о квартире, количество комнат, район, обновления последние, этажность.

В целом, фильтр в парсере является важным инструментом для обработки и анализа данных, позволяя управлять и выбирать только необходимую информацию для дальнейшей работы, что является достаточно хорошим плюсом при выборе данного сайта.

Основная информация после фильтрации данных, введённых пользователем выводится в столбец и в строки. Есть такие функции, как нахождение объектов по карте, также есть дополнительная функция, можно скачать фотографии с сайта без водяного знака, все показано на рисунке 1.2 *–* Главная страница*.*

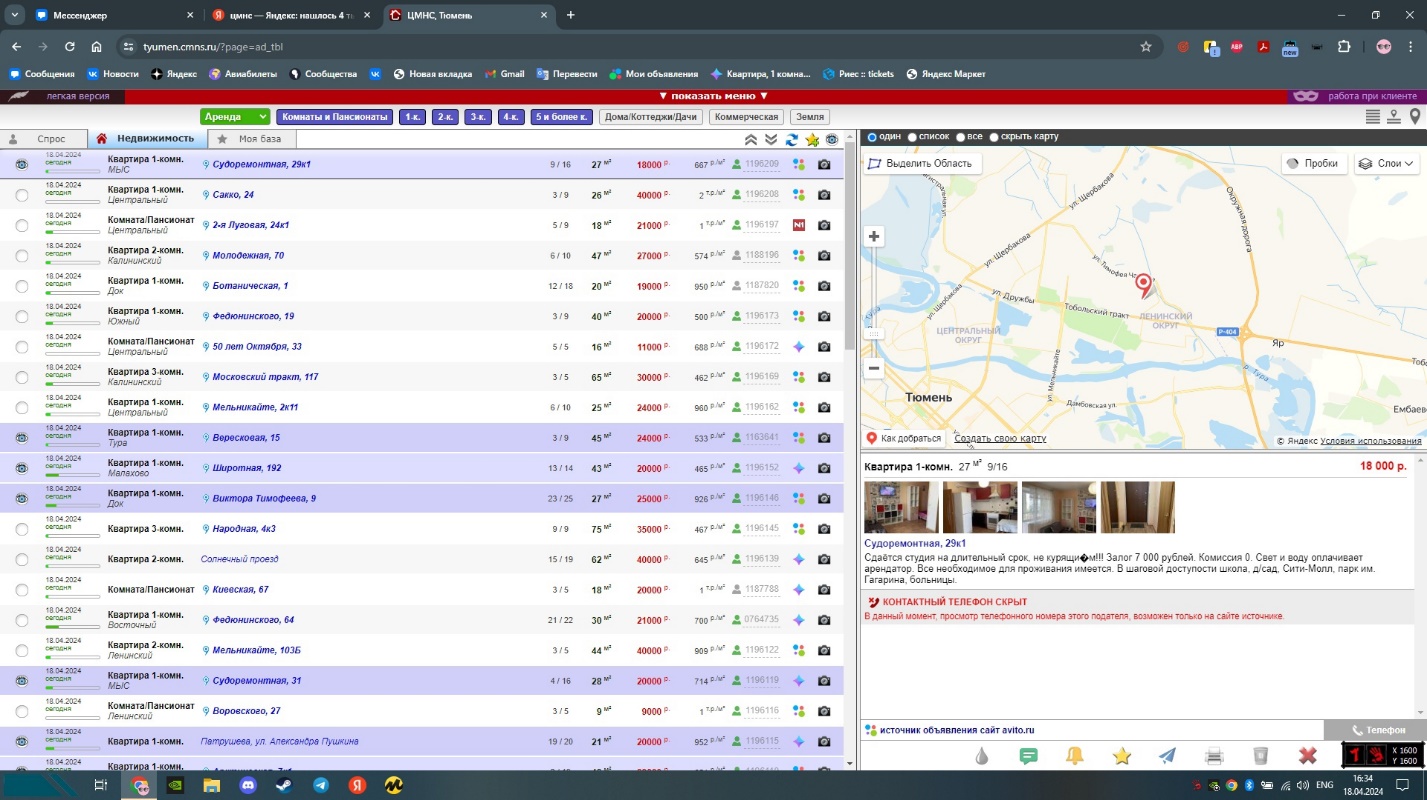


Рисунок 1.2 – Главная страница

На главной странице присутствуют адрес, площадь, этажность, стоимость, ссылка на сайт агрегатор и доступ к фотографиям без водяного знака, также есть возможность получить описание, не переходя на сторонние ссылки.

Из основного минуса, просмотренные объявления никак не помечаются, что очень плохо в риелторской сфере, так как с одного аккаунта, могут заходить несколько специалистов по недвижимости одновременно.

* + 1. **Обзор парсера «INPAS»**

Парсер «INPAS» платный сайт для парсинга объявлений с сайтов агрегаторов, сайт имеет доступный интерфейс и работает в онлайн режиме. Программа берёт оплату почасового использования, что является большим минусом при использование данного парсера.

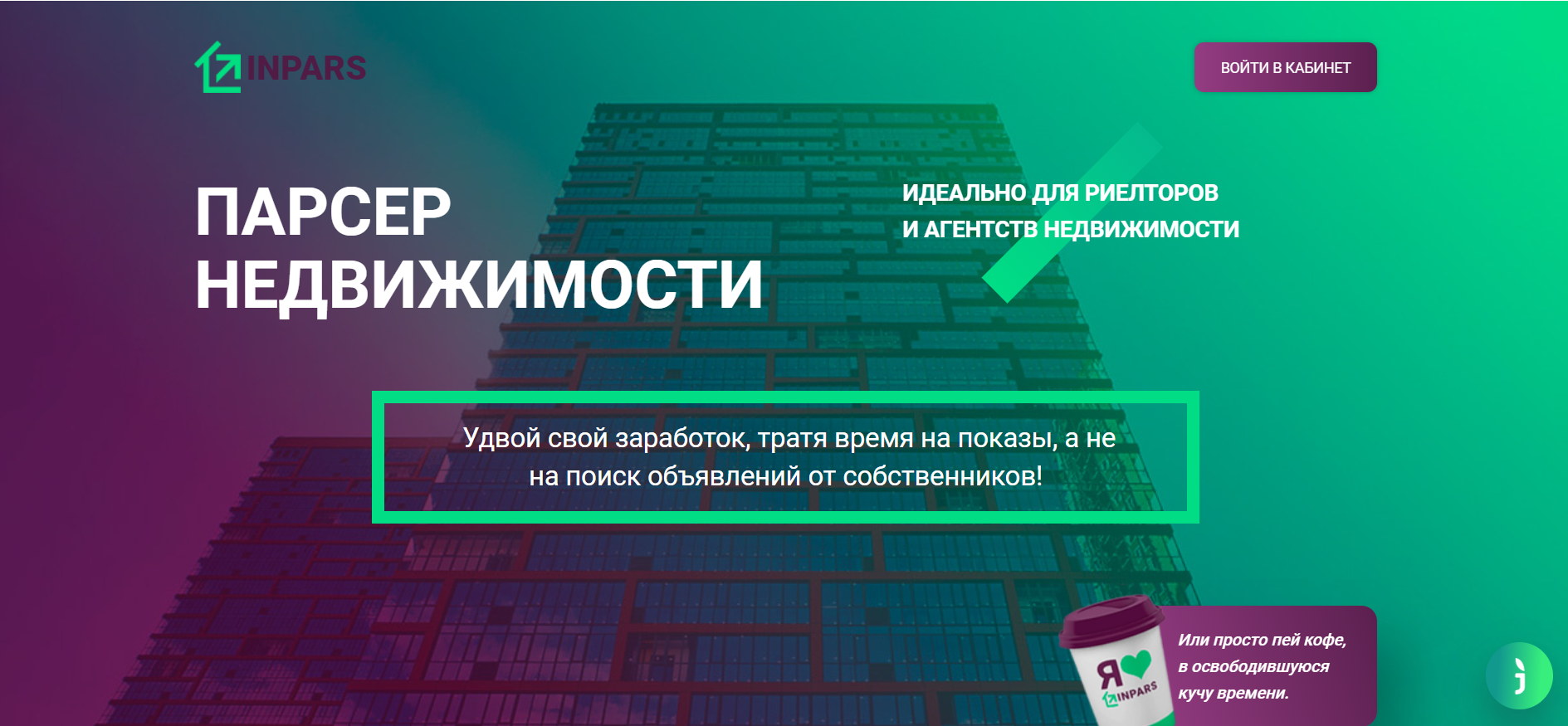


Рисунок 1.3 – Главная страница сайта

На сайте есть фильтр, где можно отсортировать информацию, разделяется по выбору нужной потребности, такой как (купить, снять , сдать, продать). Возможен выбор объявлений с фото, избранных объявлений ранее сохранившим пользователем, скрытые объявления, объявления по которым не дозвонились собственнику. Более подробно можно рассмотреть фильтр на рисунке 1.4 – Фильтр «INPAS».

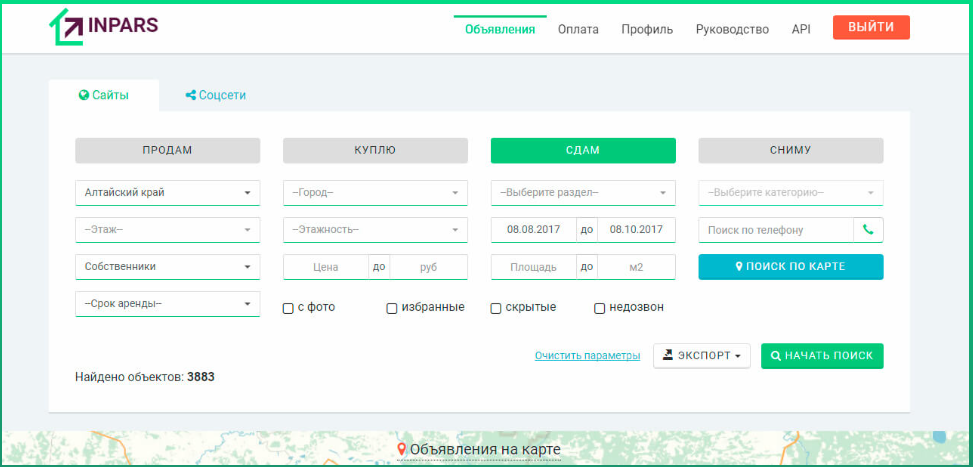


Рисунок 1.4 – Фильтр "INPAS"

* 1. **Перспективы создания парсера для агентства недвижимости**

Парсер в сфере недвижимости становится все более востребованным инструментом для получения конкурентного преимущества и автоматизации процессов.

Недвижимость относится к числу тех вложений, которые приносят значительную ценность для бизнеса при минимальных рисках. Однако, отрасль недвижимости очень конкурентоспособна, так как в каждом регионе есть сотни риэлторов. К счастью, решения, ориентированные на данные, с помощью парсинга сайтов, позволяют компаниям находить важные идеи для разработки бизнес-стратегий с более высокой доходностью.

Специалисты по недвижимости используют информацию с парсера для структурирования списков объектов недвижимости. Индустрия недвижимости динамично меняется и компаниям на рынке требуются инновационные стратегии, чтобы опережать конкурентов. Таким образом, с помощью парсинга сайтов, риэлторы могут оценить потенциальных покупателей и информацию о продажах и сдаче квартир. Кроме того, сбор информации с сайтов недвижимости – незаменимый метод поиска достоверных и актуальных знаний, поскольку он обеспечивает практическую полезность в повседневной деятельности компании. Компании, занимающиеся недвижимостью, могут собирать данные по каждому объекту недвижимости и анализировать полученную информацию.

Преимущества парсинга для индустрии недвижимости:

* Получение стратегической информации.

Парсинг поможет получить информацию, в результате анализа которой можно принимать различные бизнес-решения. Большие объемы данных можно сохранить в предпочтительном формате и хранить в базе данных для использования в будущем. Парсинг поможет легко получить данные, расположенные в труднодоступных для анализа ресурсах (сложная структура на сайте или данные в приложении), обеспечивая огромный массив информации, необходимый для того, чтобы опережать конкурентов. Использование обновляемых данных по недвижимости – эффективный способ опередить конкурентов в отношении изменений на рынке.

* Увеличение эффективности аренды.

Уровень дохода от аренды – это важная вещь, которую необходимо учитывать перед инвестированием в недвижимость. Инвесторы могут легко определить, в какую недвижимость инвестировать, на основе собранной информации. Более того, анализ различных сайтов дает ответы на насущные вопросы, например, о том, какие объекты недвижимости имеют наибольшую рентабельность инвестиций в определенных областях. Работа с данными – это также эффективный способ удостовериться, что информация, размещенная на сайте риэлтора, является исчерпывающей и прозрачной, чтобы поддерживать конкурентное преимущество.

* Принятие инвестиционных решений.

Получение точных и обновленных данных о недвижимости – сложный процесс, который часто приводит к тому, что инвестор принимает слепые финансовые решения. С помощью парсинга инвесторы в сфере недвижимости могут получить доступ к эмпирическим, применимым и качественным данным. Информация помогает инвесторам принимать обоснованные решения, а не полагаться на неполные или устаревшие данные. Сбор данных о недвижимости с различных сайтов, перечисляющих недвижимость, является необходимостью для инвестиционного исследования.

* Отслеживание стоимости недвижимости.

Сервисы парсинга недвижимости предоставляют информацию, используемую для отслеживания стоимости различных объектов. Например, домовладельцы могут отказаться от Интернета, чтобы установить стоимость аналогичной недвижимости. Таким образом, желающие продать могут установить справедливые цены, чтобы привлечь потенциальных покупателей, но при этом обеспечить им прибыльную сделку.

1. **Проектирование парсера для агентства недвижимости** 
   1. **Концепция парсера для агентства недвижимости**

Парсер позволяет быстро и эффективно собирать информацию о доступных объектах недвижимости (квартиры, дома, земельные участки и т. д.), и предоставлять ее агентству для анализа и работы с клиентами.

Концепция парсера для агентства недвижимости, которая предоставлена в данной работе:

Настройка и выбор источников данных: Информация будет переноситься с популярных сайтов агрегаторов Авито, Циан.

Разработка алгоритма парсинга: Алгоритм обходит выбранные источники данных, извлекать необходимую информацию (цену, площадь, адрес, этаж, дату, когда опубликовали объявление) и сохраняет ее в удобном формате в виде таблицы.

Обработка данных и фильтрация: После сбора данных их необходимо обработать и отфильтровать по заданным критериям фильтрация нужна для того, чтобы предоставить агенту только актуальные и подходящие варианты по его запросу, в фильтре содержится такая информация, как количество комнат, адрес, площадь, цена, этаж, автор публикации (это может быть как частное лицо, так и агентство недвижимости).

Визуализация данных: для удобства работы агентов недвижимости, данные о доступных объектах недвижимости будут визуализированы в виде списка с фильтрами, что поможет быстро находить нужные варианты для клиентов.

Автоматизация процесса: для повышения эффективности работы агентов недвижимости, парсер будет настроен на автоматическое обновление данных с выбранных источников.

Таким образом, парсер для агентства недвижимости поможет значительно сократить время на поиск и анализ доступных объектов недвижимости, улучшить качество обслуживания клиентов и увеличить эффективность работы агентства.

* 1. **Концепция интерфейса**

Одним из основных преимуществ приложения является наличие интерфейса, который удобен и прост в использовании. Проанализировав конкурентные решения, был выявлен ряд недочетов, который был учтен при разработке концепции интерфейса.

Для качества работы в виде основных цветов данной программы были выбраны черный, белый и синий, такие цвета являются классическими и универсальными, что делает их популярным выбором для дизайна программ. Черный цвет может добавить элегантности и сдержанности, белый – простоты и чистоты, а синий – профессионализм, спокойствие . Кроме того, черно-белая цветовая гамма обеспечивает хороший контраст и читаемость текста, что делает информацию на экране легко воспринимаемой для пользователей.

При первом открытии веб-сайта мы попадаем на форму входа, где можно как войти, так и зарегистрироваться на сайте, для комфортной работы.

Вход при разработке интерфейса необходим для изучения потребностей пользователей и определения их ожиданий от продукта. Это позволяет создать удобный и интуитивно понятный интерфейс, который будет соответствовать потребностям целевой аудитории. Также вход при разработке интерфейса позволяет определить основные функциональные требования, бизнес-цели и бюджет проекта. В результате правильно выполненного входа можно создать уникальный и успешный продукт, который будет приносить пользу пользователям. (рисунок 2.1).

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 2.1 – Форма входа

Регистрация при разработке интерфейса нужна для того, чтобы пользователи могли создавать учетные записи и входить в систему для доступа к функционалу, сохранению персональных настроек, и сохранению данных.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 2.2 – Форма регистрации

При входе открывается главная страница, на которой изображены фильтры, разделения на квартиры, загородные дома, участки. Также есть возможность фильтрации и оставление комментариев (рисунок 2.3)

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 2.3 – Главная страница

Фильтры при разработке парсера для агентства недвижимости необходимы для оптимизации и улучшения процесса поиска и сортировки данных. Они позволяют выбирать определенные критерии для поиска недвижимости, такие как тип объекта, цена, район, количество комнат и другие характеристики.

Использование фильтров позволяет пользователям быстро и удобно находить желаемый объект недвижимости, а также сортировать результаты поиска в соответствии с их предпочтениями. Это делает процесс поиска более эффективным и удобным для пользователей агентства недвижимости.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2.4 – Фильтр

Разработка комментариев – это улучшение пользовательского опыта. Учитывать отзывы пользователей при проектировании интерфейса парсера, чтобы сделать его более удобным и интуитивно понятным для пользователей.

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 2.5 – Комментарии

## **2.3 Структура базы данных**

База данных для парсера агентства недвижимости необходима для хранения и организации информации о доступных объектах недвижимости, таких как квартиры, дома, земельные участки и коммерческие помещения. В базе данных хранятся различные данные о каждом объекте, такие как адрес, описание, характеристики, цена, фотографии и контактная информация продавца или агентства, все это продемонстрировано на Рисунке 2.6

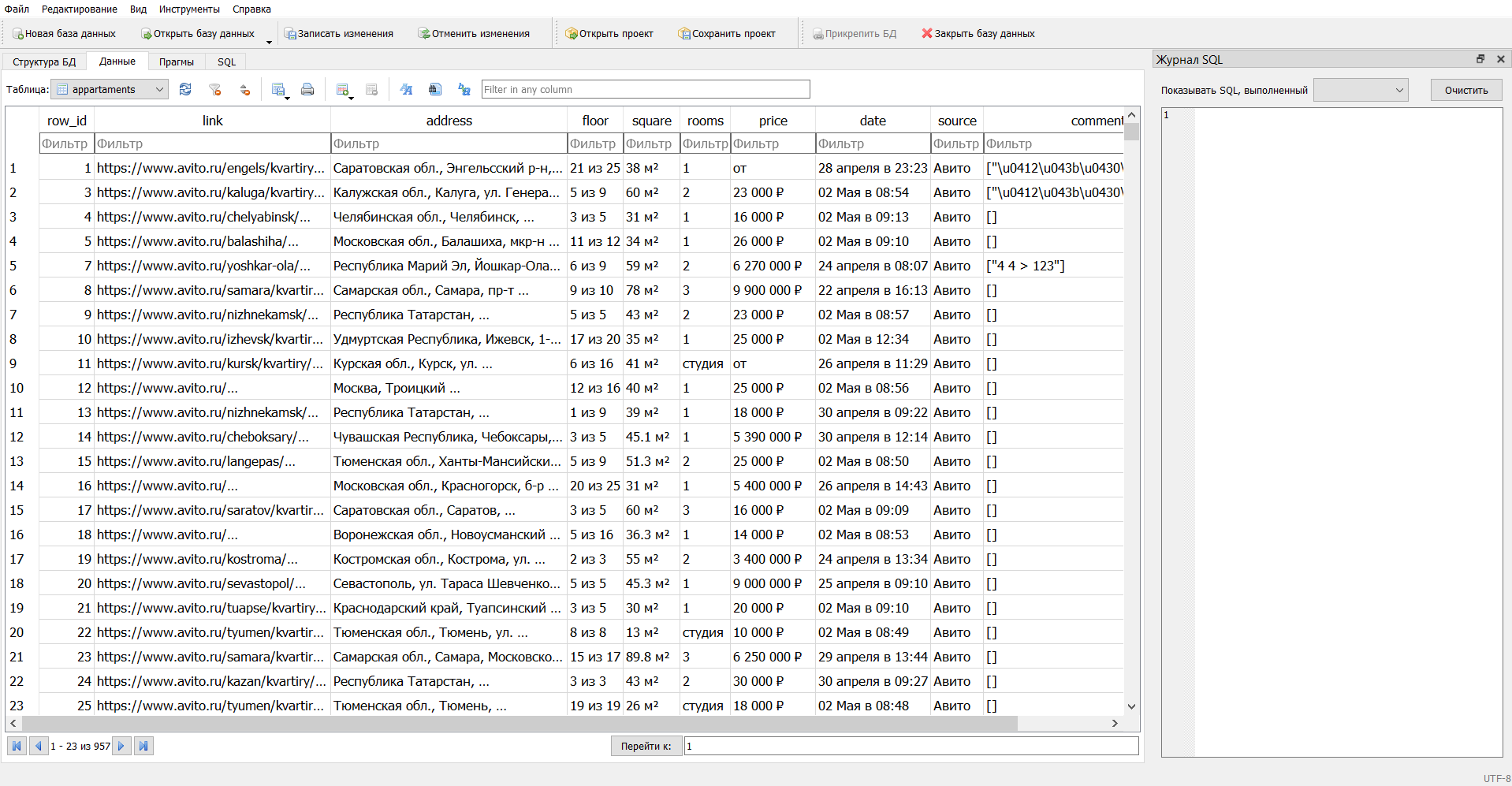


Рисунок 2.5 – База данных парсинга

Благодаря базе данных парсер агентства недвижимости может эффективно обрабатывать и анализировать информацию о новых и обновленных объектах недвижимости, обновлять информацию на сайте, а также предоставлять пользователям возможность фильтрации и поиска объектов по различным критериям.

База данных с зарегистрированными людьми является важным инструментом для повышения эффективности работы администратора в парсере для агентства недвижимости и обеспечения качественного обслуживания клиентов указана на Рисунке 2.6

Изображение выглядит как текст, снимок экрана, дисплей, диаграмма

Автоматически созданное описание

Рисунок 2.5 – База данных администратора

База данных зарегистрированных клиентов может быть использована для настройки целевых рекламных кампаний, отправки персонализированных предложений и акций, что поможет привлечь новых клиентов и удержать текущих.

1. **Разработка парсера для агентства недвижимости**
   1. **Выбор языка программирования и инструментов**

PyCharm *–* это среда программирования для языка Python, или IDE. Средами называют программы, в которых можно писать, запускать и отлаживать код, устанавливать новые расширения и дополнительные модули. Это мощный многофункциональный инструмент для разработчиков.

Вот некоторые основные характеристики PyCharm:

* редактор кода:

PyCharm имеет мощный редактор кода с подсветкой синтаксиса, автодополнением, переходом к определению функций или классов и многими другими полезными функциями;

* отладка:

в IDE есть инструменты для устранения ошибок Python-кода, которые позволяют устанавливать точки останова, следить за значениями переменных во время выполнения и анализировать стек вызовов;

* управление проектами:

PyCharm позволяет создавать и управлять проектами Python, облегчая работу с файлами, пакетами и зависимостями;

* виртуальные окружения:

для изоляции зависимостей проектов PyCharm поддерживает виртуальные окружения, что позволяет легко управлять версиями пакетов для каждого проекта;

* интеграция с устройствами контроля версий:

PyCharm поддерживает устройства контроля версий, такие как Git, Mercurial, Subversion и другие, что упрощает совместную работу над проектами;

* инструменты анализа кода:

IDE имеет инструменты для статического анализа кода, поиска потенциальных ошибок, оптимизации импортов и других полезных функций;

* поддержка web-разработки:

PyCharm обладает хорошей интеграцией с веб-технологиями, такими как HTML, CSS, JavaScript и Django.

Использование PyCharm предоставляет ряд преимуществ по сравнению с другими средствами разработки. Плюсы использования данной IDE для программирования:

* мощные инструменты подсказок и автодополнения, которые позволяют увеличить производительность и качество кода;
* интегрированная система рефакторинга позволяет легко изменять структуру кода, делая его более читаемым и понятным;
* отличная интеграция с инструментами управления версиями, такими как Git, что упрощает работу в команде;
* интуитивно понятный интерфейс, который упрощает навигацию по проекту;
* широкий выбор плагинов для настройки среды под свои потребности;
* интеллектуальный анализ кода поможет выявить потенциальные проблемы и предложить варианты их решения;
* возможность использования отладчика для пошагового анализа выполнения кода;
* поддержка различных фреймворков, библиотек и инструментов для анализа кода.

Все эти особенности делают PyCharm незаменимым инструментом для разработчиков на Python, помогая им создавать высококачественное программное обеспечение быстро и эффективно.

Python *–* универсальный язык, на котором можно писать практически всё. Фреймворк Django *–* один из инструментов для его эффективного применения в веб-разработке.

Flet – это новый современный фреймворк, позволяющий создавать мультиплатформенные веб-, мобильные и настольные приложения на Python. Flet позволяет упаковывать ваши приложения для распространения на всех платформах - Windows, macOS, Linux, Web, iOS (ipa), Android (apk/aab).

Основные особенности Flet Python:

* простота использования:

фреймворк предоставляет интуитивно понятный синтаксис и легко настраиваемую структуру проекта, что делает процесс разработки более гибким и эффективным;

* мощные функциональные возможности:

flet Python включает в себя широкий набор модулей и библиотек для работы с HTTP-запросами, базами данных, аутентификацией и многими другими возможностями;

* гибкость и расширяемость:

разработчики могут легко расширять функциональность фреймворка, добавляя собственные модули и плагины, а также интегрировать сторонние библиотеки и сервисы;

* поддержка асинхронных операций:

flet Python обеспечивает поддержку асинхронного программирования, что позволяет создавать быстрые и отзывчивые веб-приложения;

* встроенная защита от уязвимостей:

фреймворк предоставляет механизмы для защиты приложений от распространенных видов атак, таких как SQL-инъекции, кросс-сайтовый скриптинг и другие.

Плюсы фреймворка Flet Python:

* простота использования:

flet Python предоставляет простой интерфейс для создания веб-приложений, что делает его идеальным выбором для разработчиков начального уровня;

* масштабируемость:

flet Python обладает возможностью масштабироваться и поддерживать различные типы веб-приложений, от простых блогов до сложных корпоративных систем;

* поддержка Python:

фреймворк построен на языке программирования Python, что делает его доступным для широкого круга разработчиков, знакомых с этим языком.

Минусы фреймворка Flet Python:

* недостаточная документация:

некоторые пользователи могут столкнуться с проблемами из-за недостаточной документации и ограниченной поддержки сообщества;

* ограниченные возможности:

по сравнению с другими популярными фреймворками, Flet Python может иметь ограниченные возможности и функциональность;

* популярность:

flet Python не так популярен, как некоторые другие фреймворки, такие как Django или Flask, что может создавать проблемы при поиске ресурсов и поддержки.

* 1. **Написание функционального кода страниц**

Основное назначение этого класса *–* обеспечение взаимодействия с базой данных (запись и чтение данных) с использованием многопоточности (блокировка `Lock`).

Подключение базы данных к парсингу:  
lass DB:

def \_\_init\_\_(self, path):

super(DB, self).\_\_init\_\_()

self.\_\_lock = Lock()

self.\_\_db\_path = path

self.\_\_cursor = None

self.\_\_db = None

self.init()

def init(self):

if not os.path.exists(self.\_\_db\_path):

self.\_\_db = sqlite3.connect(self.\_\_db\_path, check\_same\_thread=False)

self.\_\_cursor = self.\_\_db.cursor()

self.\_\_cursor.execute('''

CREATE TABLE users(

row\_id INTEGER primary key autoincrement not null,

name TEXT,

lastname TEXT,

email TEXT,

login TEXT,

password TEXT

)

''')

self.\_\_cursor.execute('''

CREATE TABLE appartaments(

row\_id INTEGER primary key autoincrement not null,

link TEXT,

address TEXT,

floor TEXT,

square TEXT,

rooms TEXT,

price TEXT,

date TEXT,

source TEXT,

comments TEXT

)

''')

self.\_\_db.commit()

else:

self.\_\_db = sqlite3.connect(self.\_\_db\_path, check\_same\_thread=False)

self.\_\_cursor = self.\_\_db.cursor()

def db\_write(self, queri, args):

with self.\_\_lock:

self.\_\_cursor.execute(queri, args)

self.\_\_db.commit()

def db\_read(self, queri, args):

with self.\_\_lock:

self.\_\_cursor.execute(queri, args)

return self.\_\_cursor.fetchall()

Подключение базы данных к парсеру для агентства недвижимости будет полезным для автоматизации процесса сбора и обновления информации о доступных объектах недвижимости. С помощью парсера и базы данных агентство сможет проводить более эффективный мониторинг рынка недвижимости, следить за изменениями в ценах и доступности объектов, а также получать уведомления о новых предложениях.

Данный код, который будет предоставлен ниже представляет класс UI, который представляет пользовательский интерфейс приложения.

class UI:

def \_\_init\_\_(self, config, db):

super(UI, self).\_\_init\_\_()

self.\_\_vault\_keys = ['current\_user']

self.\_\_config = config

self.\_\_db = db

self.\_\_login = Login(db)

self.\_\_pagemain = MainPage(db)

self.\_\_reg = Registration(db)

def main(self, page: ft.Page):

flet\_navigator = VirtualFletNavigator(

{

'/': self.\_\_login.login,

'main': self.\_\_pagemain.main\_page,

'registration': self.\_\_reg.registration

}

)

flet\_navigator.render(page)

Виртуальный навигатор render вызывается для отображения соответствующей страницы пользователю.

Ниже код представляет класс Login, который используется для выполнения операций входа в систему. Метод login проверяет введенные пользователем логин и пароль.

class Login:

def init(self, db):

super(Login, self).init()

self.\_\_crud = CRUD(db)

def login(self, pg: PageData):

def login\_btn(e):

if len(user\_login.value) > 0 and len(user\_password.value)

if self.\_\_crud.check\_login(user\_login.value, user\_password.value):

pg.page.session.set('creds', [user\_login.value, user\_password.value])

pg.navigator.navigate('main', pg.page)

else:

open\_dlg\_modal(None)

else:

open\_dlg\_modal(None)

def close\_dlg(e):

dlg\_modal.open = False

pg.page.update()

def redirect\_on\_reg(e):

pg.navigator.navigate('registration', pg.page)

def open\_dlg\_modal(e):

pg.page.dialog = dlg\_modal

dlg\_modal.open = True

pg.page.update()

Ниже код представляет класс Registration, который отвечает за процесс регистрации пользователей. Этот класс обеспечивает функционал для регистрации новых пользователей и взаимодействия с базой данных для проверки и добавления учетных данных пользователей.

class Registration:

def \_\_init\_\_(self, db):

super(Registration, self).\_\_init\_\_()

self.\_\_crud = CRUD(db)

self.\_\_dlg\_modal = None

def registration(self, pg: PageData):

def reg\_btn(e):

if not self.\_\_crud.check\_login(login\_field.value, pass\_field.value):

self.\_\_crud.add\_user([name\_field.value, sur\_name\_field.value, email\_field.value, login\_field.value, pass\_field.value])

pg.page.session.set('creds', [login\_field.value, pass\_field.value])

pg.navigator.navigate('main', pg.page)

else:

open\_dlg\_modal(None)

def close\_dlg(e):

dlg\_modal.open = False

pg.page.update()

def open\_dlg\_modal(e):

pg.page.dialog = dlg\_modal

dlg\_modal.open = True

pg.page.update()

def validate(event):

print([name\_field.value, sur\_name\_field.value, email\_field.value, login\_field.value, pass\_field.value])

if all([name\_field.value, sur\_name\_field.value, email\_field.value, login\_field.value, pass\_field.value]):

reg\_button.disabled = False

else:

reg\_button.disabled = True

pg.page.update()

Ниже код представляет собой метод, который парсит информацию о квартирах с веб-страницы и сохраняет ее в базу данных.

Этот код выполняет парсинг информации о квартирах с веб-страницы и сохраняет ее в базу данных для дальнейшего использования.

match index:

case 0:

# парсим адрес

data.append(page\_data.split('\n')[0])

case 1:

# парсим этаж, площадь, количество комнат

page\_data\_array = page\_data.split('\n')

for g in ['Этаж', 'Общая площадь', 'Количество комнат']:

flag = False

for search in page\_data\_array:

if g in search:

flag = True

data.append(search[len(g)+2:])

if not flag:

data.append('-')

case 2:

# парсим цену

data.append(page\_data.split('\n')[0])

case 3:

# парсим дату публикации

if 'сегодня' in page\_data or 'вчера' in page\_data:

current\_date = datetime.now()

# Форматируем дату в требуемом формате

formatted\_date = "{:02d} {}".format(current\_date.day, self.\_\_months[current\_date.month])

v\_index = page\_data.find(' в ')

data.append(f'{formatted\_date} {page\_data[v\_index+1:]}')

else:

data.append(page\_data[2:])

data.append('Авито')

# Проверяем отсутствие ошибки

if len(set(data)) > 3:

self.\_\_crud.add\_apartment(data)

Фильтрация данных

dlg\_modal = ft.AlertDialog(

modal=True,

title=ft.Text("Фильтр"),

content=ft.Column(

[

ft.Column(

[

ft.Row(

[

ft.Text(value='Комнаты:', size=12),

ft.FilledButton(text='не выбрано', on\_click=studio\_btns) ft.FilledButton(text='Студия', on\_click=studio\_btns), ft.FilledButton(text="1", on\_click=studio\_btns

ft.FilledButton(text="2", on\_click=studio\_btns),

ft.FilledButton(text="3", on\_click=studio\_btns),

ft.FilledButton(text="4+", on\_click=studio\_btns)

]

),

ft.Row([

ft.Text(value='Автор', size=12)

ft.FilledButton(text="Частное лицо", on\_click=author),

ft.FilledButton(text="Риелтор", on\_click=author)

])

]

),

ft.Column(

self.\_\_filter\_fields

)

]

),

actions=[

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной курсовой работы были успешно достигнуты поставленные цели и выполнены все поставленные задачи.

В рамках работы был разработан парсер для агентства недвижимости. Для достижения данной цели были решены следующие задачи:

* проанализирована предметная область;
* разработана концепция парсера для агентства недвижимости;
* разработан дизайн парсера для агентства недвижимости;
* разработана функциональная и интерфейсная часть парсера для агентства недвижимости;
* проведено тестирование и оптимизация разработанного парсера для агентства недвижимости.

Проведенный анализ предметной области позволил определить требования к разрабатываемому инструменту, а разработка концепции парсера и его дизайна обеспечили эффективность и удобство использования.

Разработанная функциональная и интерфейсная часть парсера была успешно протестирована и оптимизирована, что позволило обеспечить высокую эффективность работы инструмента.

Таким образом, выполнение поставленных задач позволило создать полноценный инструмент для улучшения процессов работы агентства недвижимости и повышения качества обслуживания клиентов.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Ахрамчук, П. П. (2022). Применение сервиса Figma при проектировании интерфейса программного обеспечения.
2. Гайнутдинов, В. Р. (2023). ОБЗОР ФРЕЙМВОРКОВ PYTHON ДЛЯ ВЕБ-РАЗРАБОТКИ. ЛУЧШАЯ НАУЧНАЯ СТАТЬЯ 2023, 6-10.
3. Денисенко, В. В., Парфенов, А. К., Маврин, В. В., & Скрыпников, А. А. (2022). ПАРСИНГ ДАННЫХ.
4. Златопольский, Д. (2022). Основы программирования на языке Python. Litres.
5. Лозовой, М. В. (2022). ПРИМЕНЕНИЯ FIGMA-ОНЛАЙН-СЕРВИСА ДЛЯ РАЗРАБОТКИ ИНТЕРФЕЙСОВ. In Инжиниринг предприятий и управление знаниями (ИП&УЗ-2022) (pp. 115-119).
6. Постолит, А. (2021). Python, PyCharm и Django для начинающих. БХВ-Петербург.
7. Хаджаев, С. И. (2023). РАЗРАБОТКА RESTFUL API С ИСПОЛЬЗОВАНИЕМ PYTHON И DJANGO: ОСНОВЫ И ПРАКТИЧЕСКИЕ СОВЕТЫ. Статья описывает процесс разработки RESTful API с использованием Python и фреймворка Django. Обсуждаются основы создания моделей данных, определения URL-адресов. Research and implementation.
8. Чистяков, П. А. (2020). ПАРСИНГ КАК СПОСОБ СБОРА ДАННЫХ ИЗ ОТКРЫТЫХ ИСТОЧНИКОВ. In Новые информационные технологии в научных исследованиях (pp. 113-115).
9. Python: особенности и перспективы [Электронный ресурс]. Режим доступа: https://gb.ru/blog/python/ – (дата обращения: 25.11.2024).
10. Python подробный справочник [Электронный ресурс]. Режим доступа: http://uchcom7.botik.ru/L/prog/python/python\_05.pdf – (дата обращения: 25.11.2024).