Приложение на Django для анализа качества ключевых фраз, с помощью pandas рекламной кампании яндекс Директ, выгружаемой по API

**Содержание**

1. Обзор проекта
2. Структура проекта
   1. Домашняя страница
   2. Страница «Обучение»
   3. Страница регистрации
   4. Страница полученных отчетов
3. Шаблоны страниц
   1. Домашняя страница
   2. Страница «Обучение»
   3. Страница регистрации
   4. Страница полученных отчетов
4. Примеры выгрузки в Jupiter и полученные отчеты
5. Заключение
6. Приложение 1. Пример файловой структуры проекта
7. Приложение 2. Список необходимых библиотек
8. **Обзор проекта**

Предлагается для реализации приложение для аналитика на базе фреймворка Django, которое позволит загружать данные всех рекламных кампаний по API с рекламного кабинета Яендекс Директ за определенный период времени, затем с помощью библиотеки pandas, путем сортировки, группировки и тд, получать еженедельные готовые отчеты (exel/csv формат):

* По положительной конверсии конкретной кампании:
* отчет, содержащий ключевые фразы, принесшие заявки, с низкой стоимостью просмотра (например, от 50 руб/клик и ниже)
* По отрицательной конверсии:
* отчет, содержащий ключевые фразы, не давшие конверсию, с высокой стоимостью просмотра ( от 100 руб/клик и выше )

Данные отчеты позволят ускорять еженедельную работу по обнаружению аномальных значений стоимости показов ключевых фраз, получать информацию быстро и качественно, не пропуская нужные отчеты и получая возможность корректировать рекламные кампании, уменьшая стоимость заявки. В дальнейшем возможно на базе приложения реализовать:

* выгрузку новых ключевых слов, которые не добавлены в кампанию
* автоматическую корректировку с помощью полученных отчетов
* кабинет для регистрации и возможности заказать отчет

Для реализации, потребуются:

* механизмы выгрузки информации по API из рекламного кабинета Яндекс Директ
* использование библиотеки pandas для обработки полученных таблиц
* Django для реализации интерфейса
* Bootstrap для ускорения работы со стилями

1. **Структура проекта**

Проект включает следующие ключевые компоненты:

* 2.1 **Домашняя страница**

Домашняя страница содержит меню навигации сверху с возможностью перехода на страницу "Обучение", где описан процесс получения отчета и меню навигации слева на страницу регистрации и полученных отчетов. Далее идет карточка с переходом на страницу отчета с положительной или отрицательной конверсией.

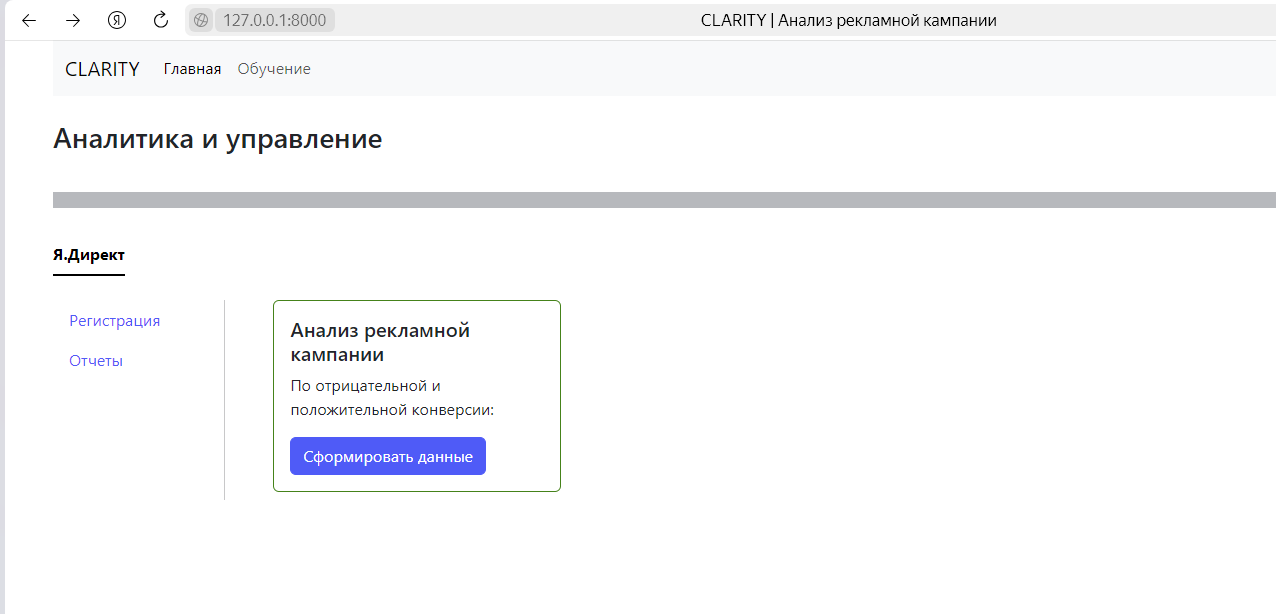


Рисунок 1 Домашняя страница

* 1. **Страница «Обучение»**

Страница содержит информацию о том, как можно сформировать отчет и что для этого понадобиться.

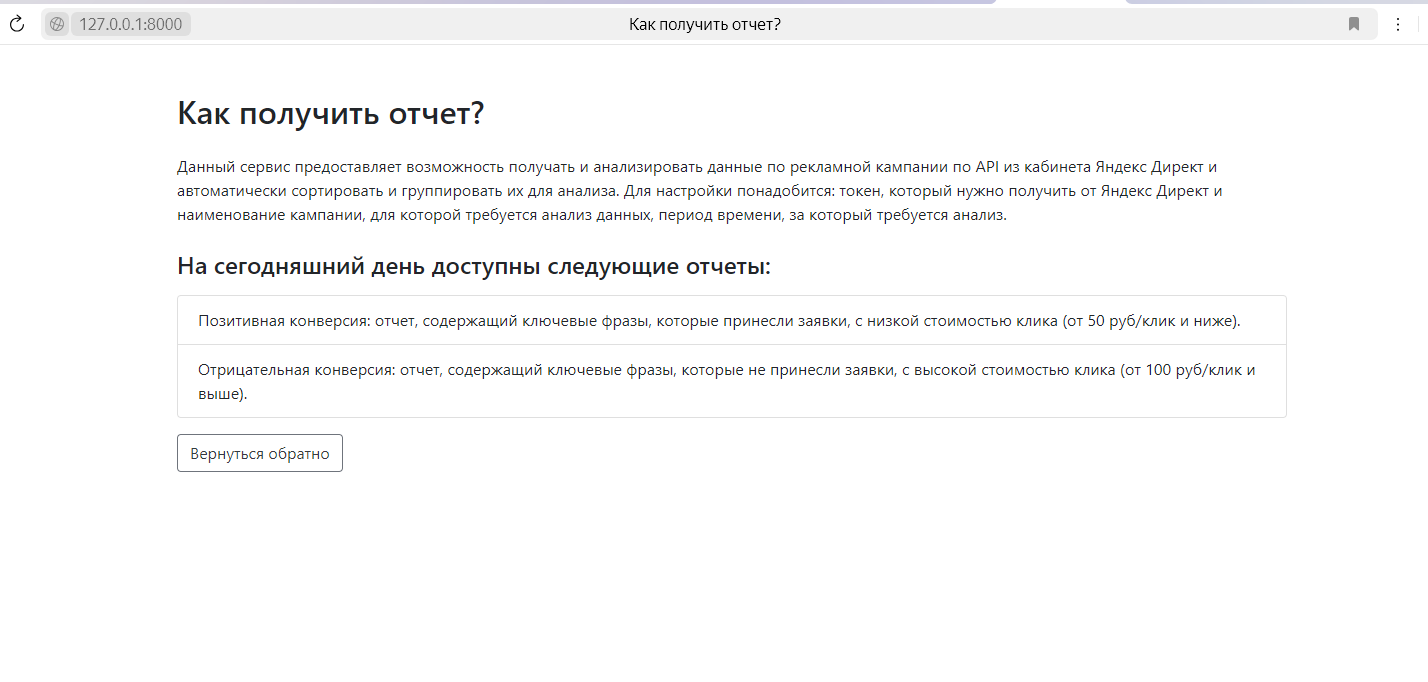


Рис. 2 Страница Обучение

* 1. **Страница регистрации**

Страница регистрации для возможности отправить данные для связи с админом и получения дальнейших инструкций.

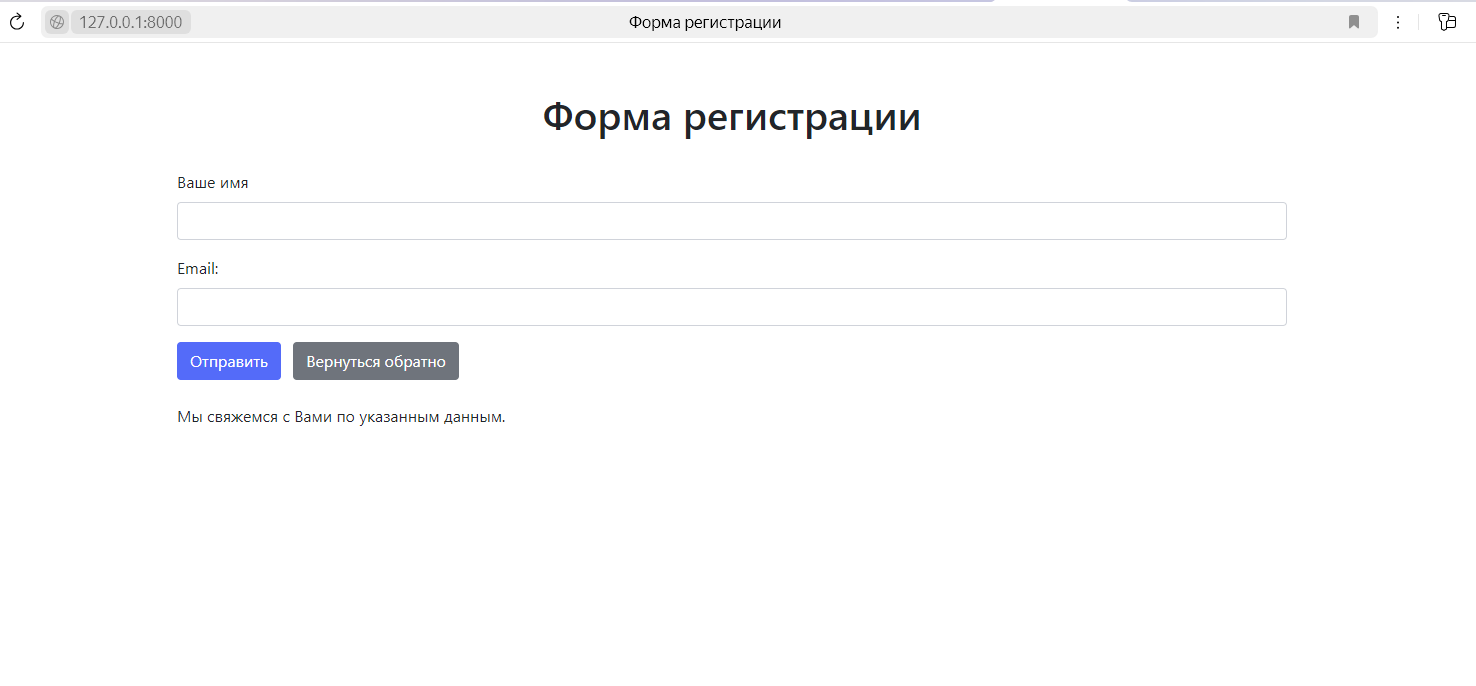


Рис.3 Страница Регистрация

* 1. **Страница полученных отчетов**

Страница отображает полученные проанализированные кампании с 2-мя отчетами, предоставленными в таблице:

По положительной конверсии конкретной кампании:

* отчет, содержащий ключевые фразы, принесшие заявки, с низкой стоимостью просмотра (например, от 50 руб/клик и ниже)

По отрицательной конверсии:

* отчет, содержащий ключевые фразы, не давшие конверсию, с высокой стоимостью просмотра ( от 100 руб/клик и выше )

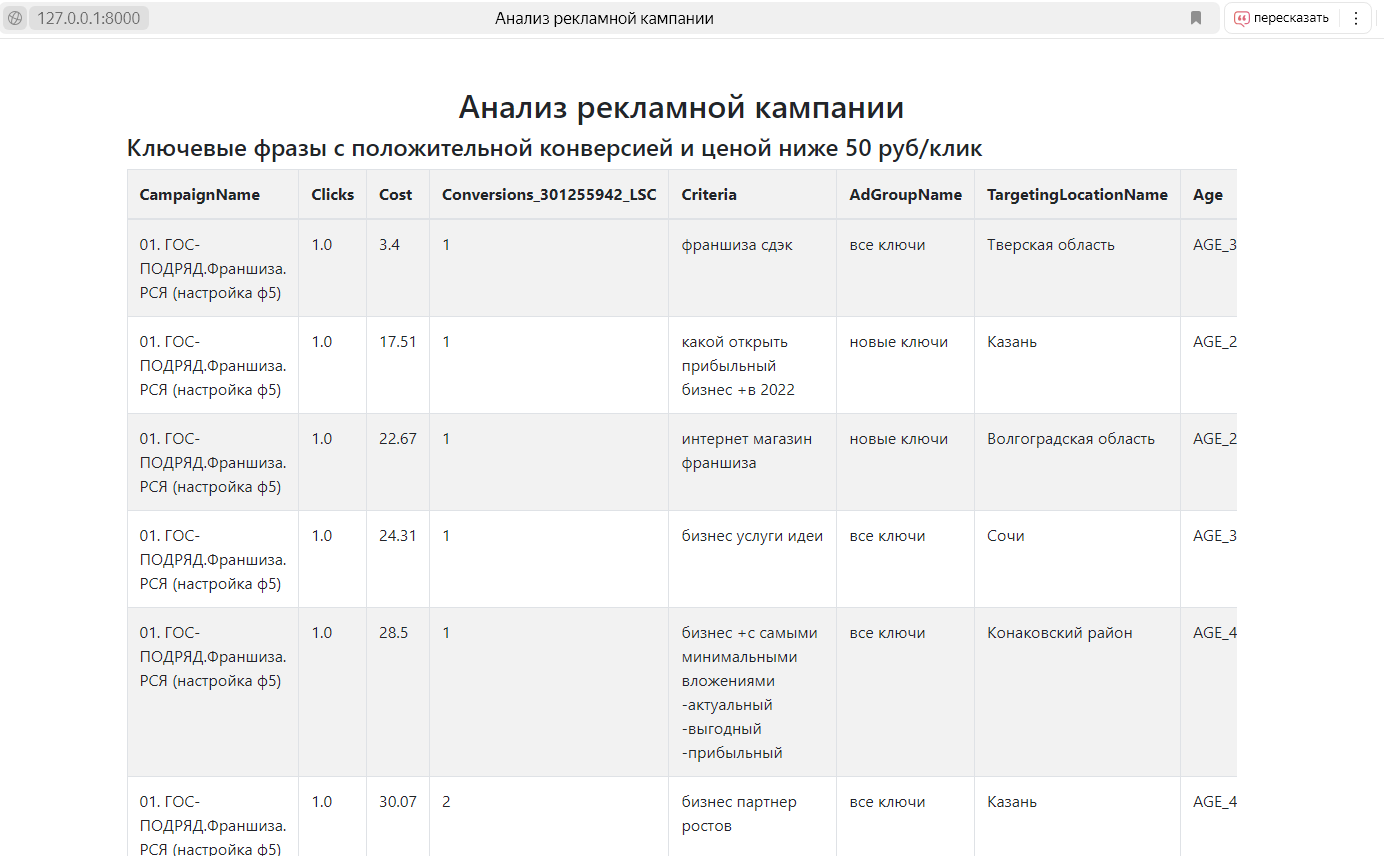
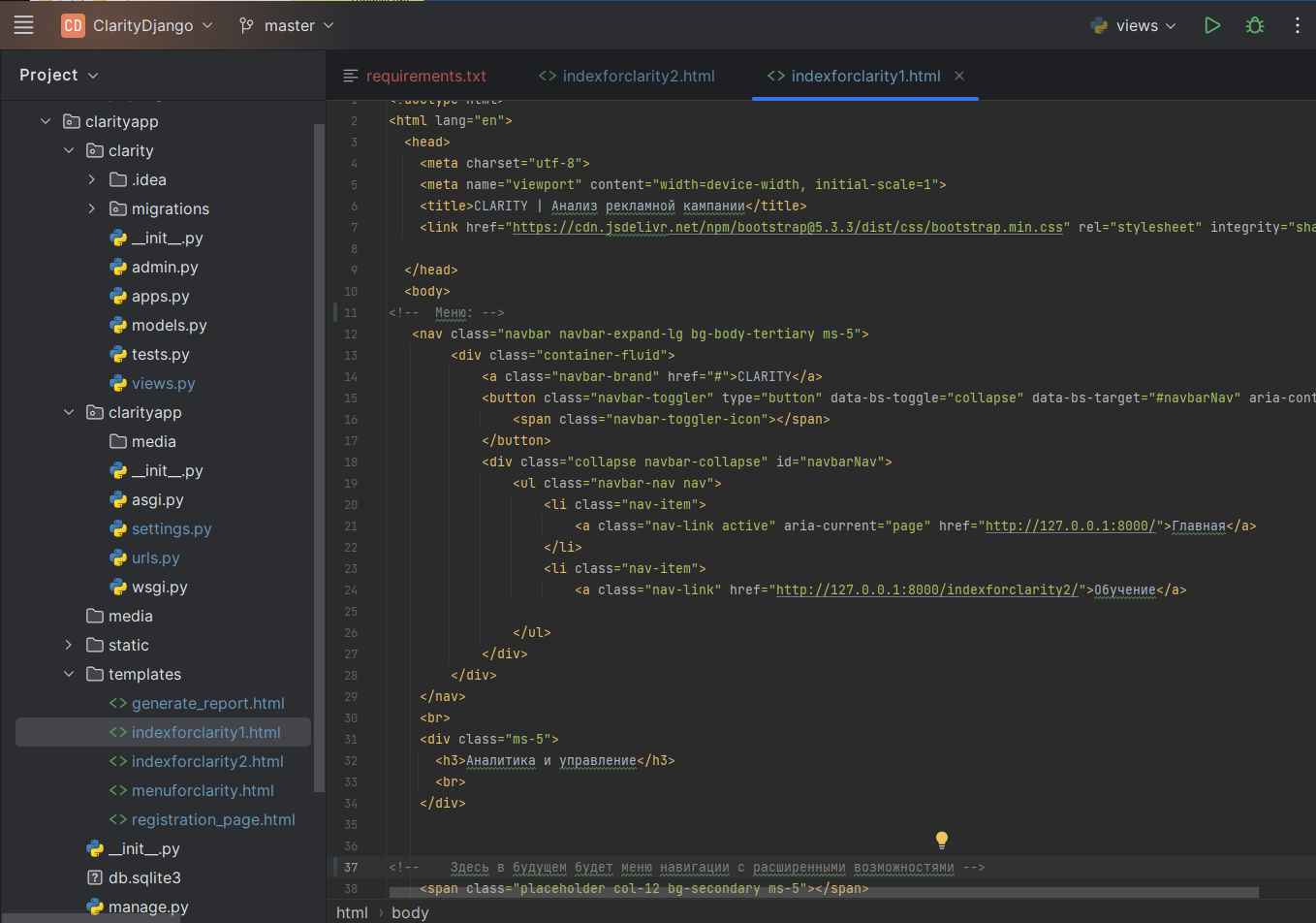


Рис.4 Страница полученных отчетов

**3.Шаблоны страниц** **и файлы**

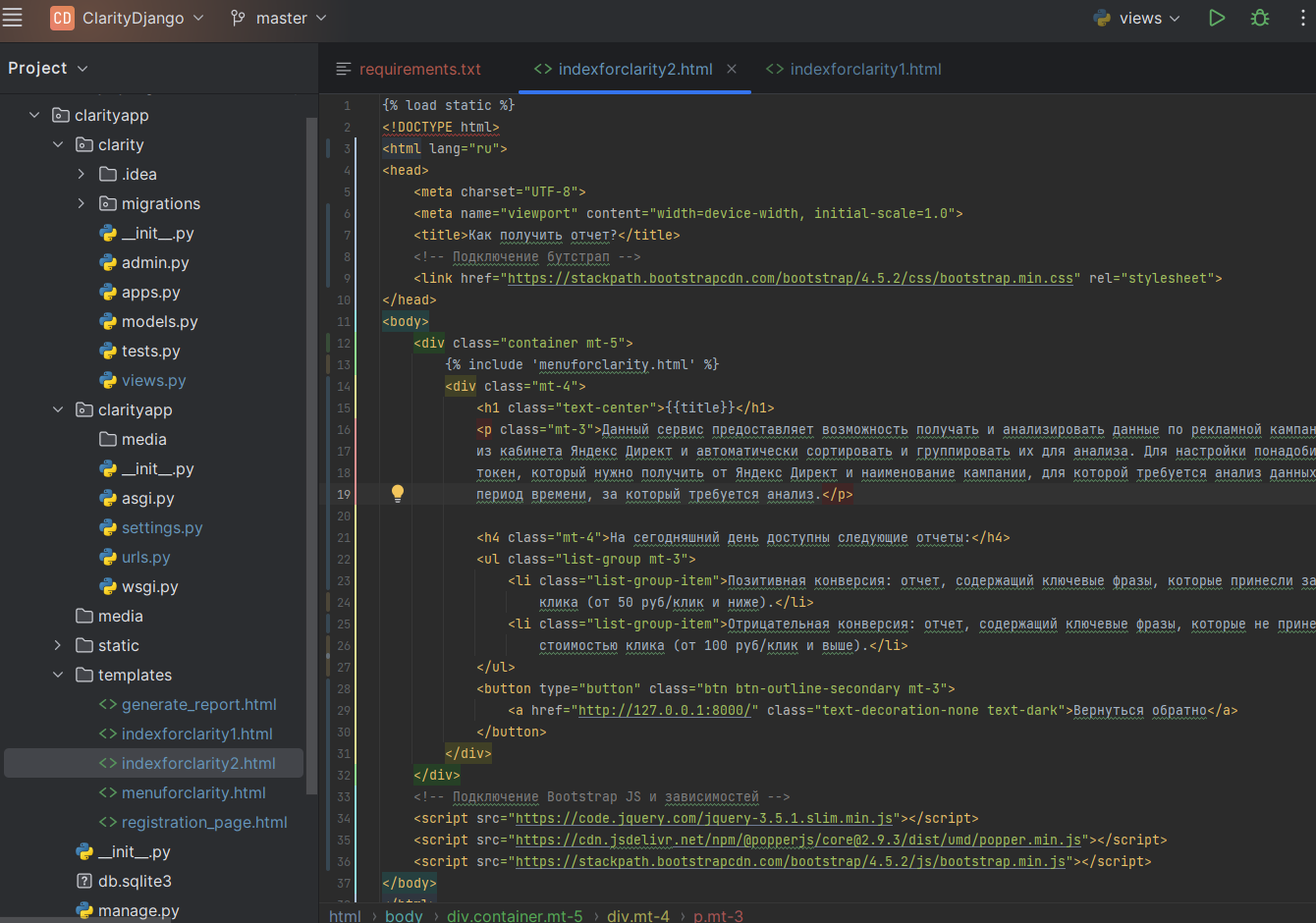
* 1. Домашняя страница

**indexforclarity1.html** – шаблон главной страницы, в нем прописана навигация и кнопки получения отчетов

****

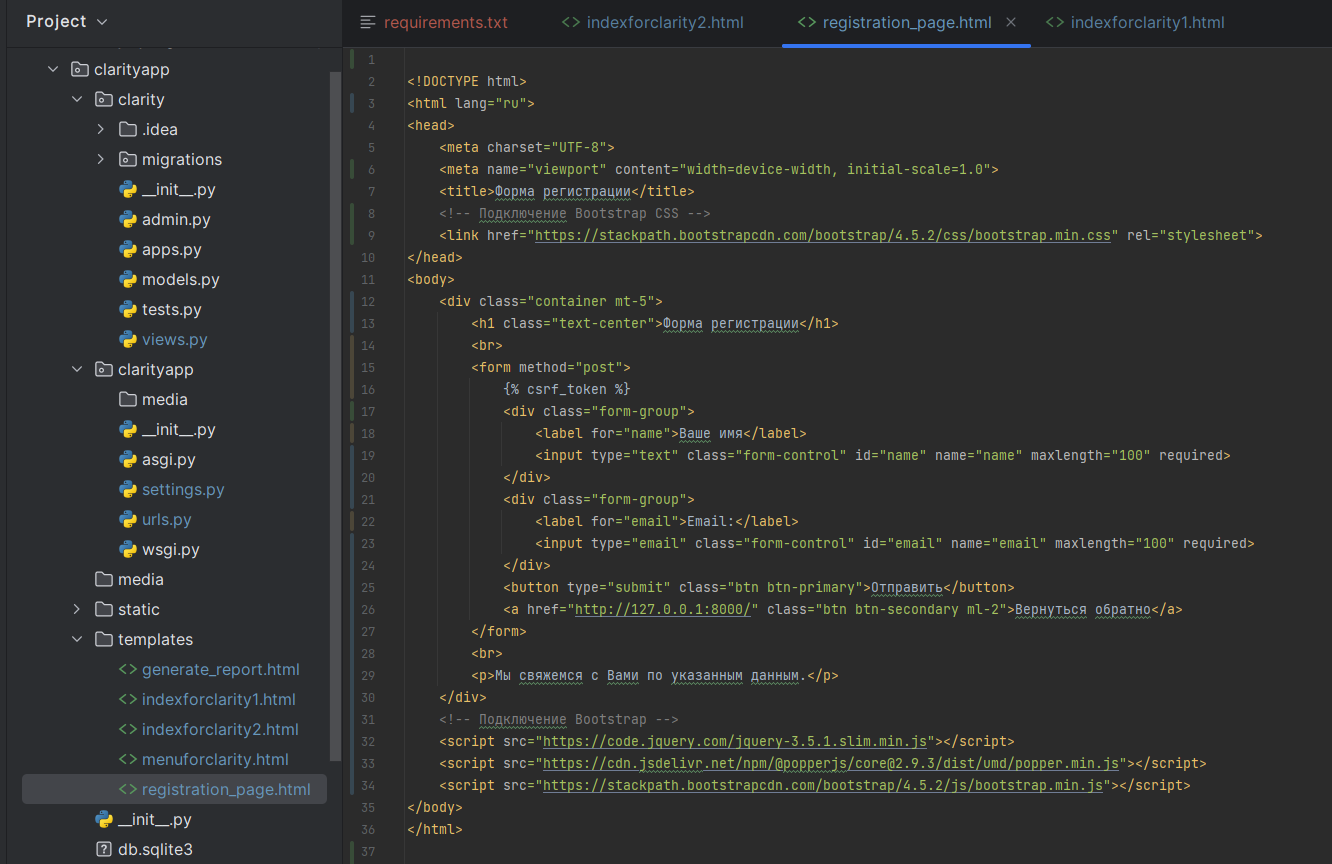
* 1. Страница «Обучение»

**Indexforclarity2.html** – шаблон второй страницы, в нем прописана информация по обучению



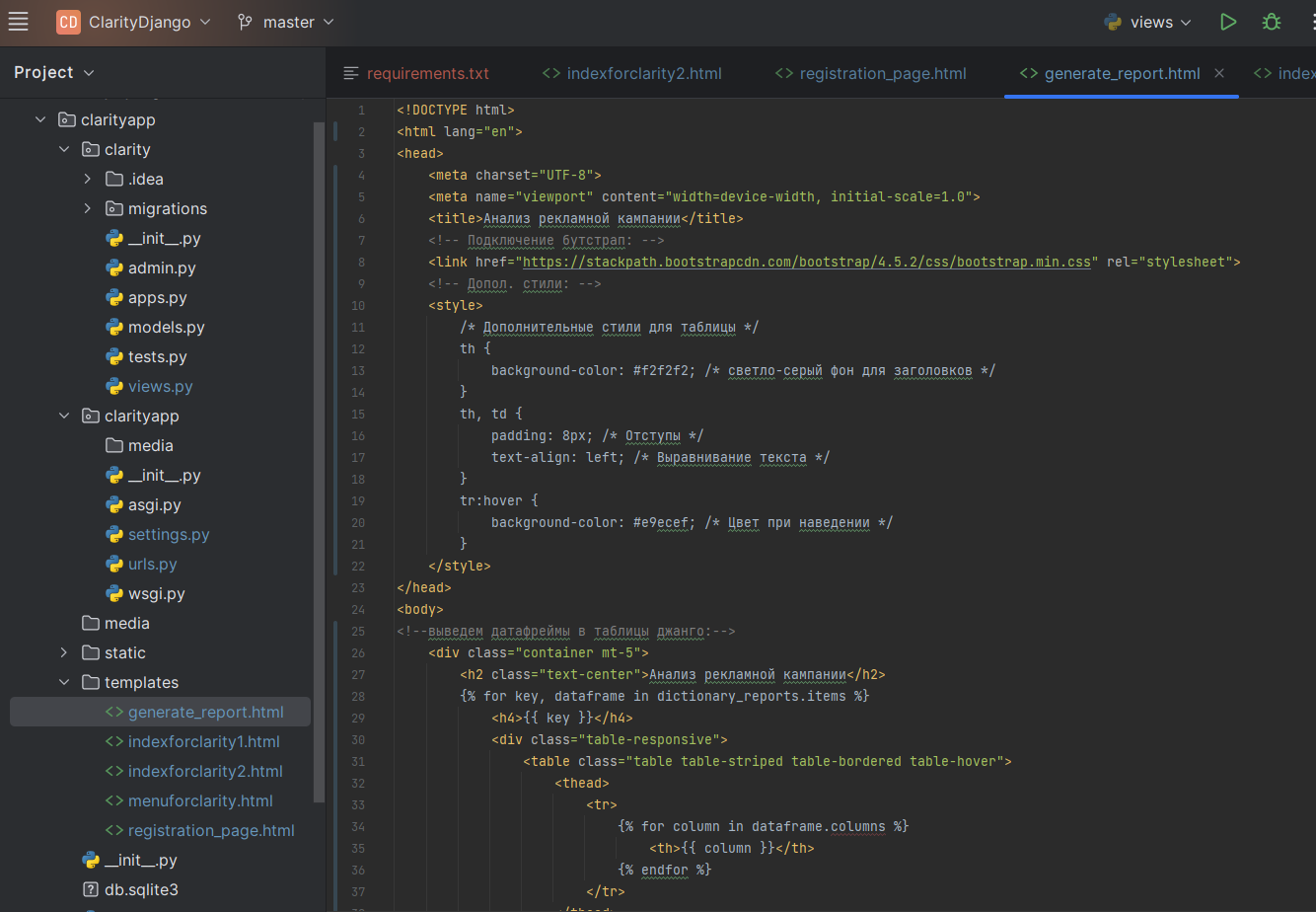
* 1. Страница регистрации

**Registration\_page.html** – шаблон страницы, где прописана регистрация пользователя



* 1. Страница полученных отчетов

**Generate\_report.html** – страница шаблон, для получения готового проанализированного отчета



1. Примеры выгрузки в Jupiter и полученные отчеты

4.1 Выгрузка по API Яндекса:

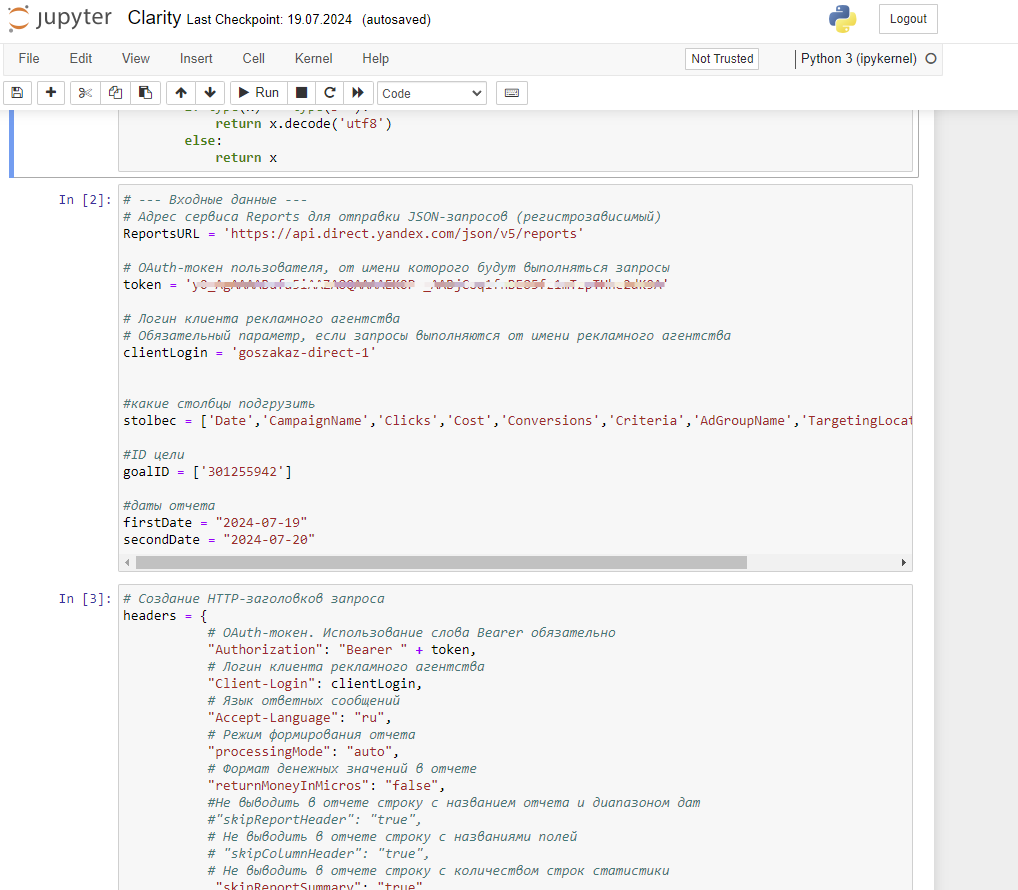


Рисунок 6. Пример выгрузки по API Яндекс Директ. Для удобства вывода и проверки работ таблиц использовался Jupiter Notebook

4.2 Пример работы с таблицами с помощью библиотеки pandas. Для удобства вывода и проверки работ таблиц использовался Jupiter Notebook

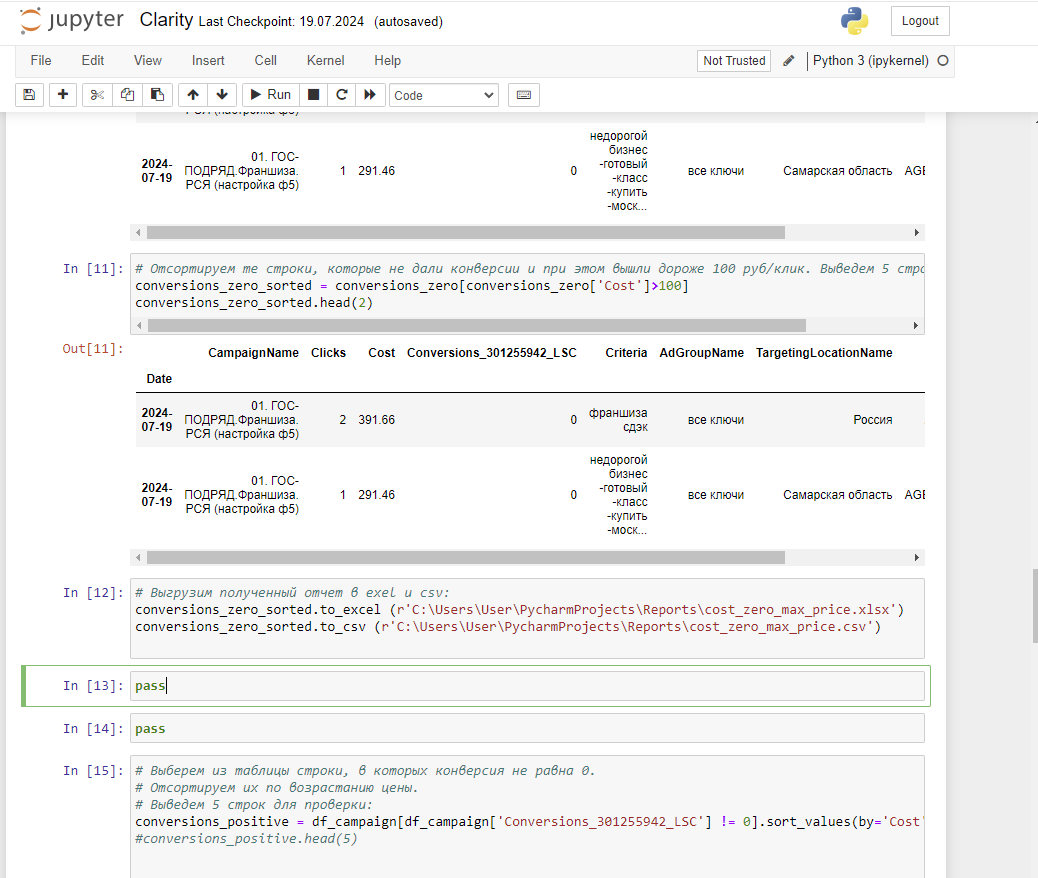


Рисунок 3 Пример работы с таблицами с помощью библиотеки pandas.

4.3 Пример полученных отчетов

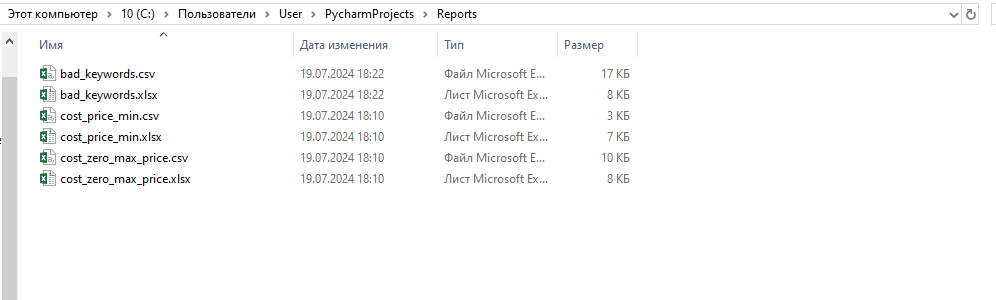


Рисунок 4 Пример сохраненных отчетов

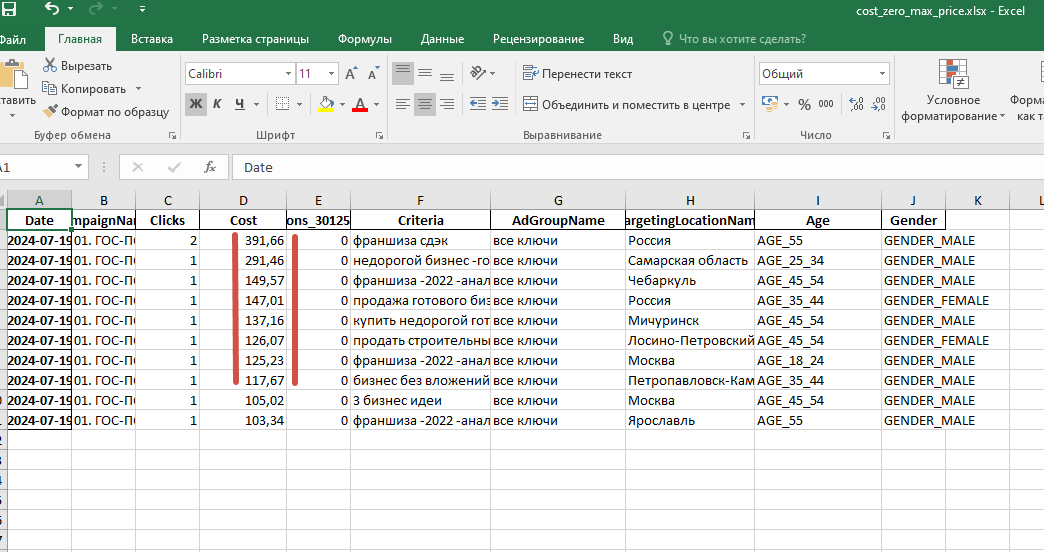


Рисунок 5 Пример полученного отчета

1. **Заключение**

Приложение предоставляет платформу для аналитика и позволяет ускорять работу по обнаружению аномальных значений стоимости показов ключевых фраз. Благодаря использованию готовых отчетов и удобного веб-интерфейса, проект возможно настроить и посадить использовать маркетолога, не имеющего опыта работы с python. Использование библиотеки pandas позволяет обрабатывать большие данные и получать их быстро.

**Приложение 1. Пример файловой структуры проекта**

│ clarity

│ migration

\_\_\_ \_\_init\_\_.py

│admin.py

│apps.py

│models.py

│tests.py

│views.py

│clarityapp

│ media

│ \_\_init\_\_.py

­­\_\_\_\_\_\_asgi.py

│requerments.txt

│settings.py

│ urls.py

│images

│media

│ static

\_\_\_\_\_\_bootstrap-5.3.3

\_\_\_\_ css

│js

│style.css

│templates

\_\_\_\_\_\_ generate\_report.html

│indexforclarity1.html

│indexforclarity2.html

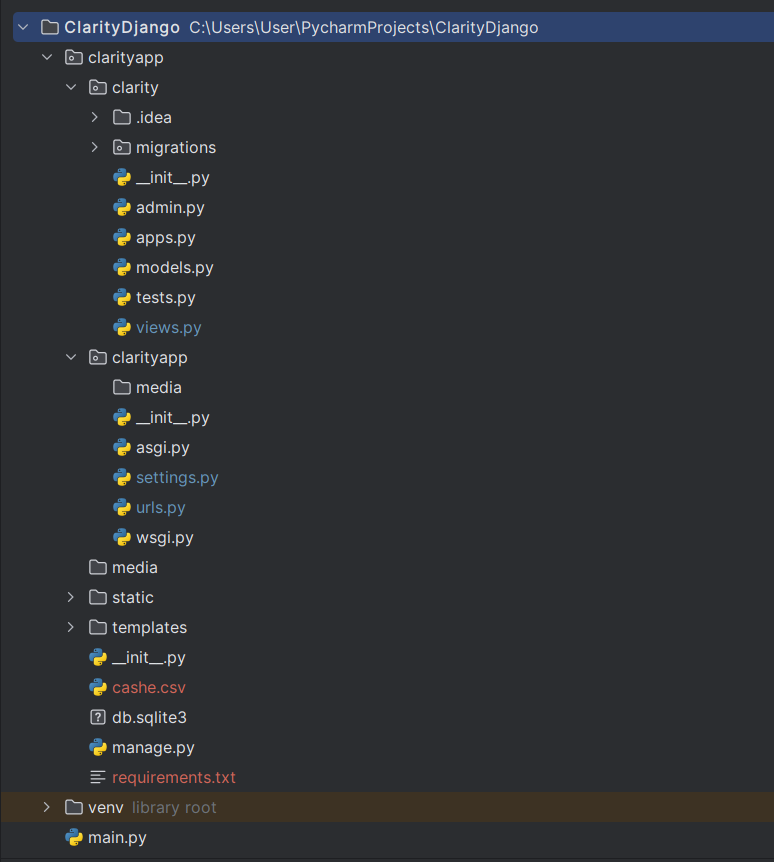
│ menuforclarity.html

│registration\_page.html

│ db.sqlite

│ manage.py

│ venv



**Приложение 2. Список необходимых библиотек**

asgiref==3.8.1  
certifi==2024.7.4  
charset-normalizer==3.3.2  
Django==5.0.7  
et-xmlfile==1.1.0  
idna==3.7  
numpy==2.0.1  
openpyxl==3.1.5  
pandas==2.2.2  
pillow==10.4.0  
python-dateutil==2.9.0.post0  
pytz==2024.1  
requests==2.32.3  
six==1.16.0  
sqlparse==0.5.1  
tzdata==2024.1  
urllib3==2.2