РАЗРАБОТКА ТЕХНОЛОГИИ СОЗДАНИЯ ФУНКЦИОНАЛЬНОГО САЙТА ДЛЯ РЕШЕНИЯ ЛОГИСТИЧЕСКИХ ЗАДАЧ НА БАЗЕ СРЕДЫ ПРОГРАММИРОВАНИЯ PYTHON И ФРЕЙМВОРКА DJANGO

Токмакова Н.М., Шкердин Д.А.

г. Орел, МБОУ СОШ № 50, 10 «А» класс

Руководитель: Демушкина О.В., г. Орел, МБОУ СОШ № 50, учитель информатики высшей квалификационной категории

В последнее время в связи с увеличением степени интеграции средств информационных технологий в разные сферы деятельности современного общества возрастают и требования к их возможностям для решения новых задач, которые предполагают большие объемы данных. Есть такие объемы информации, которые достаточно сложно обрабатывать традиционными способами.

Актуальность работы заключается в том, что с каждым годом растет потребность в изучении больших данных для российских предприятий. С этой целью все чаще используются инструменты для изучения данных. Например, язык программирования Python. Его применение используется для решения реальных задач, связанных с анализом больших объемов данных и разработкой приложений.

Цель: Создание функционального сайта для решения логистических задач на базе наиболее подходящей среды программирования.

Этапы:

I. Изучение языка программирования Python и написание программы обработки данных с его помощью.

II. Исследование фреймворка Django.

III. Создание функционального сайта.

IV. Выявление преимуществ проекта.

1. Изучение языка программирования Python и написание программы обработки данных с его помощью

Для написания программы, нами использовался язык программирования Python. Он прост в изучении, удобен в использовании. Одно из его главных преимуществ – возможность быстрого и простого подключения библиотек. В нашей работе мы предполагали работу с файлами Excel в коде программы и вывод конечного результата в таблицу. Для того чтобы связать Python и Excel в нашем проекте удобно было использовать библиотеку Openpyxl, с помощью которой происходит обращение к ячейкам, листам, и книгам MS Excel. Благодаря этой библиотеке удалось существенно облегчить написание кода программы. Суть программы довольно проста: в цикле происходит обращение к листу заданной книги, затем к ячейке листа. Со значениями ячеек производятся необходимые операции и лист перезаписывается. По результатам работы программы в конце книги создается итоговый лист с таблицей поставок.

2. Исследование фреймворка Django

Для реализации этой идеи мы посчитали более удобным выбор веб-фреймворка Django. Он отлично работает вместе с Python и позволяет создавать веб-приложения на его базе. Django появился в 2005 году. Постепенно он стал одним из лучших фреймворков, который помогает тысячам разработчиков выполнять ту или иную работу в течение нескольких минут. Изначально Django был фреймворком для языка Python, с отличным функционалом. Позднее, Django заметно упростил ряд сложностей в разработке веб приложений и придал работе более упрощенный подход.

Плюсы Django:

1) Быстрота. Django был разработан для помощи разработчикам в создании приложений настолько быстро, на сколько это возможно.

2) Полная комплектация Django работает с десятками дополнительных функций. Они помогают с аутентификацией пользователя, картами сайта, администрированием содержимого, RSS и т.д.

3) Безопасность. Работая в Django, пользователи защищены от ошибок, связанных с безопасностью и ставящих под угрозу проект. Продумана система пользовательской аутентификации.

4) Масштабируемость. Фреймворк Django наилучшим образом подходит для работы с самыми высокими трафиками.

5) Разносторонность. При помощи Django можно эффективно справляться с менеджментом контента.

3. Создание функционального сайта

Далее, мы сделали использование программы более доступным, а именно: создали функциональный сайт, который производит все операции в онлайн режиме. В роли

входных данных выступает загружаемый файл MS Excel, выходные данные – тот же файл, с уже измененным нашим сервисом содержанием.

Применение функционального сайта на практике позволяет поставщику быстро обрабатывать большие данные, необходимые в его эффективной деятельности.

В качестве примера можно привести такую аналогию: представим, что входные параметры – это продукты питания, поставляемые в определенную школу, а результат – итоговая таблица поставок.

Для того чтобы усовершенствовать программу, мы решили добавить функцию отправки email на сайт поставщика. Она запускается после обработки файла. Python содержит несколько полезных модулей, которые можно использовать для создания электронной рассылки. Для этого мы подключаем библиотеки email и smtplib. С их помощью программа производит вход на созданный нами заранее email адрес и осуществляет отправку сообщения с прикрепленным файлом. Для проверки функционирования данного сервиса мы выбрали почту одного из разработчиков.

4. Выявление преимуществ проекта

В настоящее время из доступных онлайн-программ по обработке заказов и поставок большинство являются платными, требующими регистрации и имеющими ограниченный функционал. С ними, зачастую, довольно трудно и неудобно работать. Сервис, который мы планировали разработать, должен был стать надежным и доступным для любого круга пользователей.

При работе над данным проектом были произведены следующие действия: написан блок программы, отвечающий за передачу выходного файла на почту поставщику, осуществлялась обработка действий пользователя и загрузка файла, редактирование шаблонов, форм в фреймворке Django.

В дальнейшем мы планируем продолжать работу над нашим сервисом и расширять его возможности.

Приложение 1

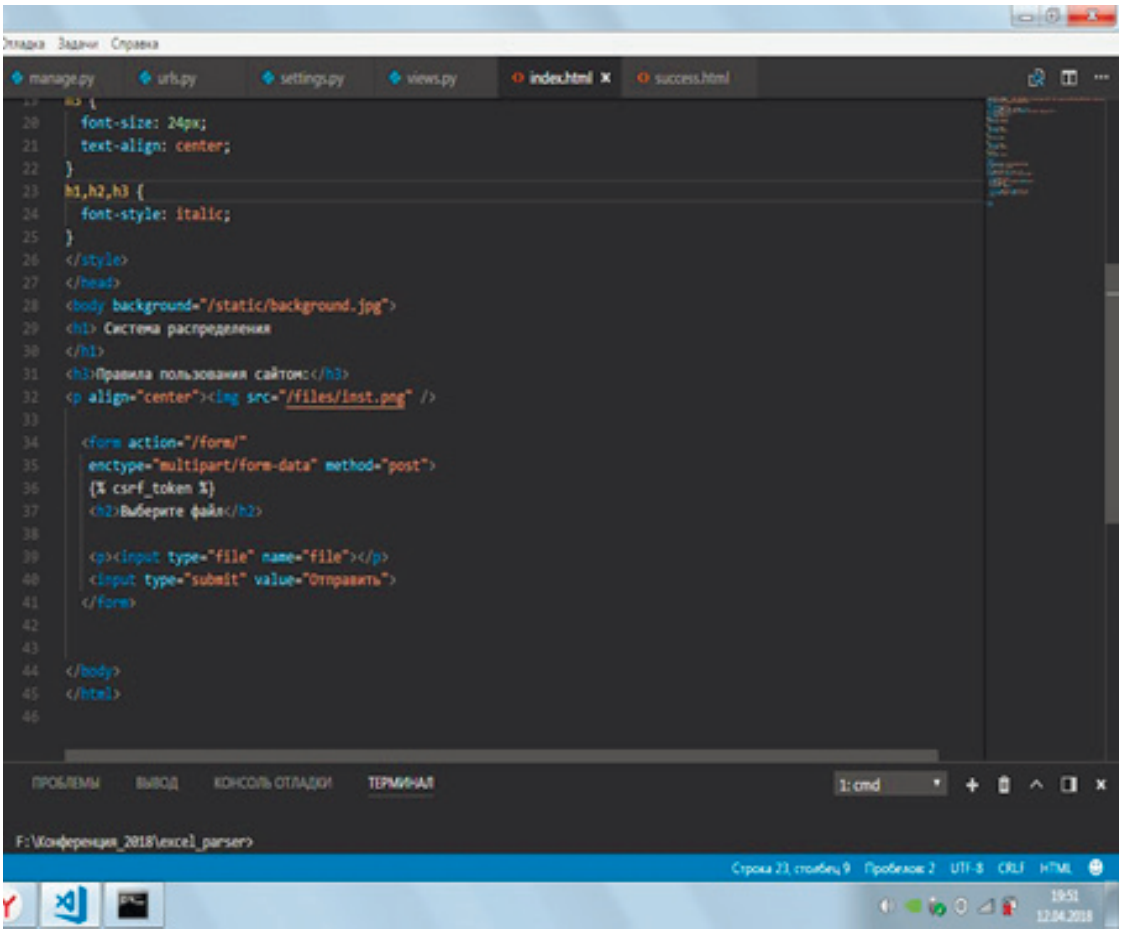

Рис. 1. Index.Основной html код

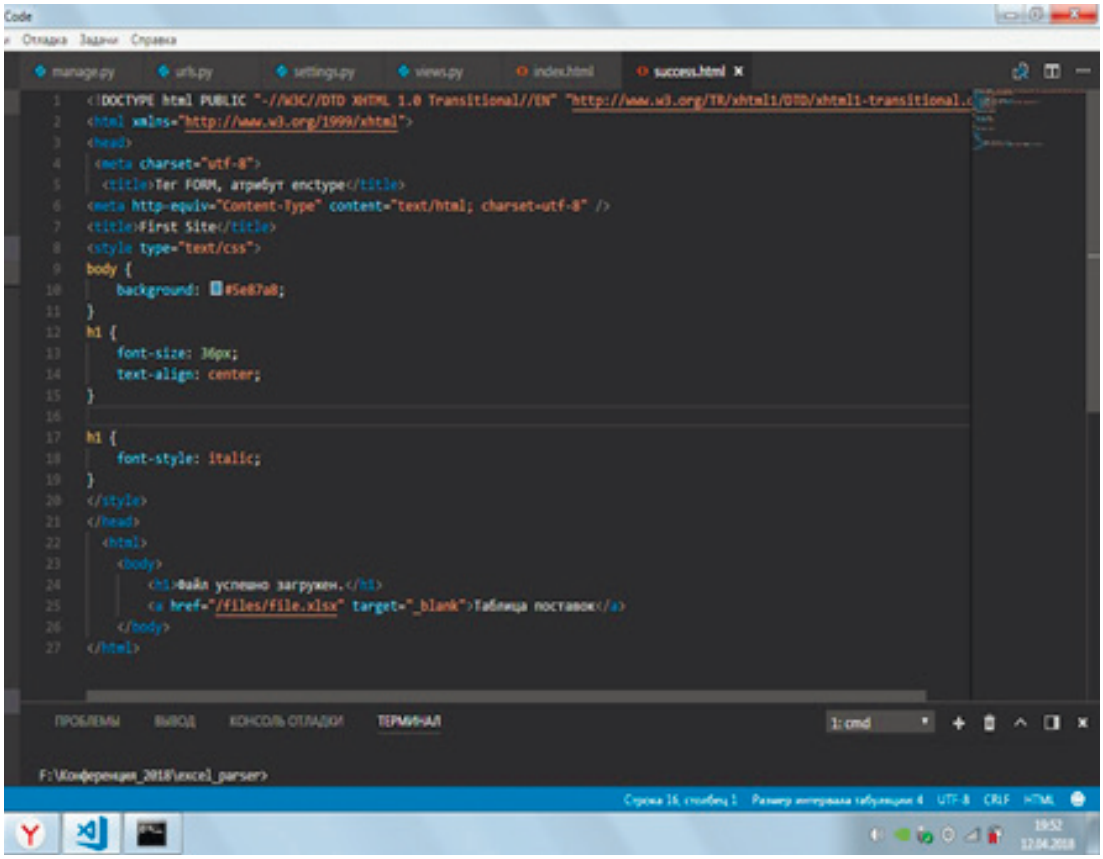

Рис. 2. Вторая страница (обновление после отправки документа)

Приложение 2

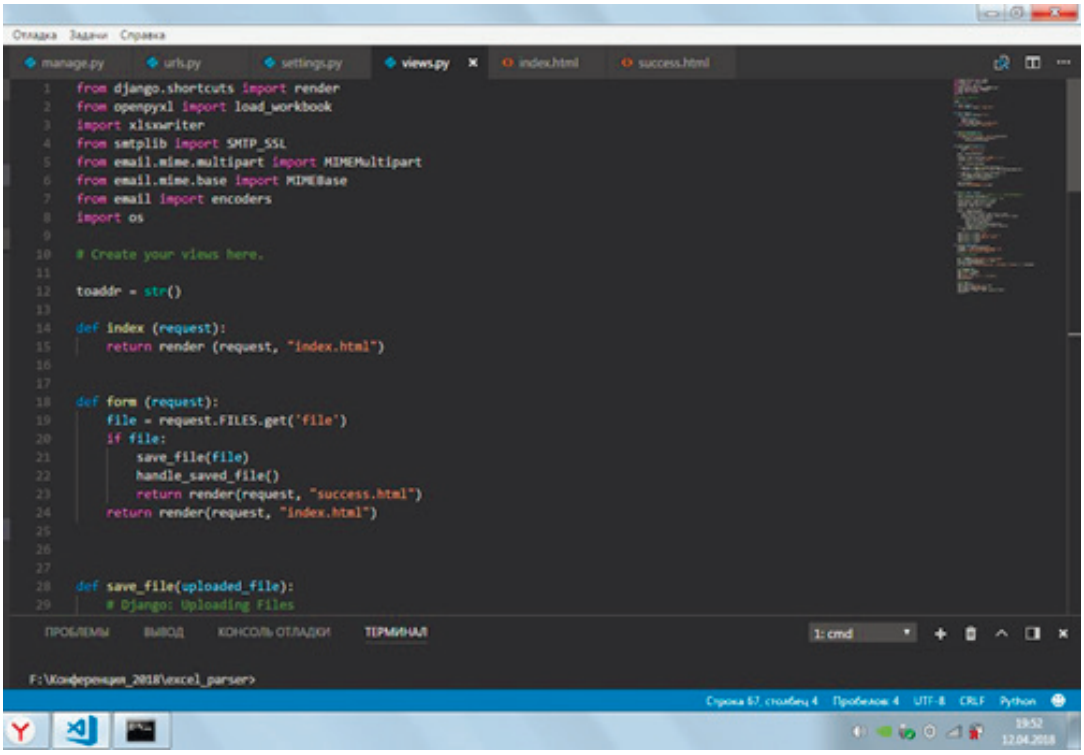

Рис. 1. Код программы на Python – Подключение библиотек

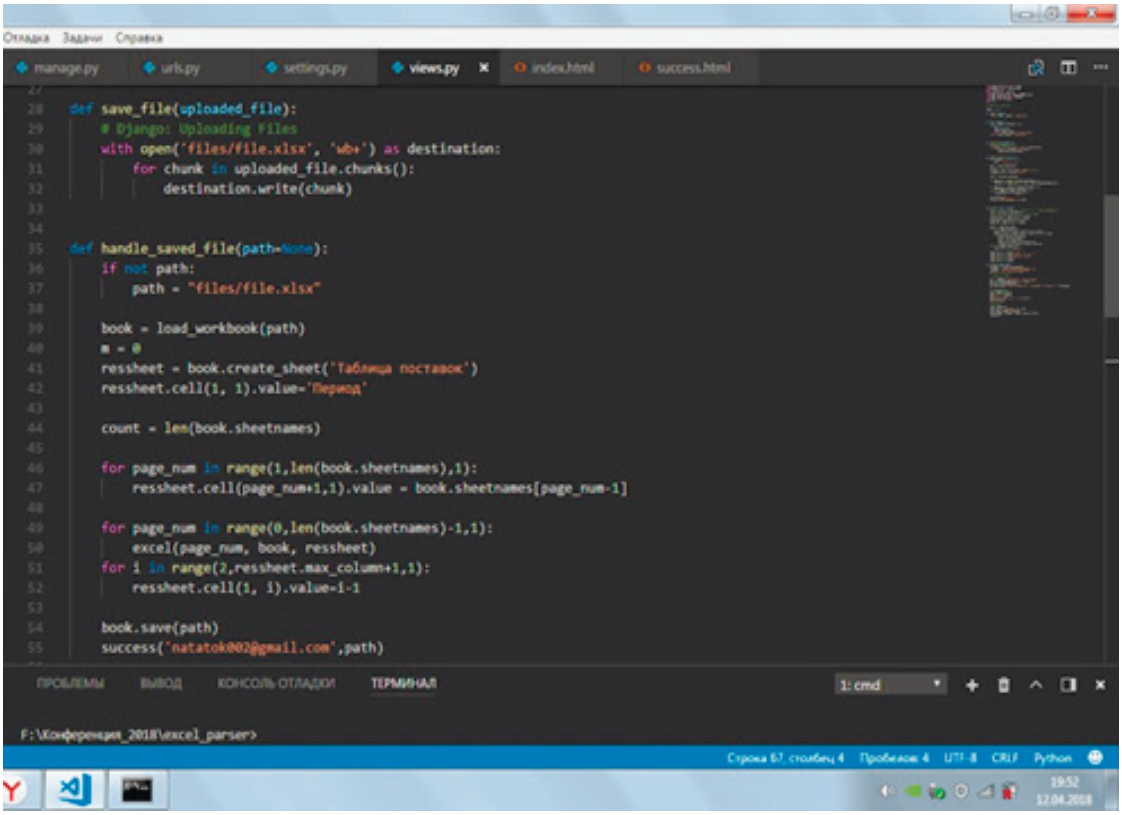

Рис. 2. Код программы на Python – Переход между страницами и загрузка файла

Приложение 3

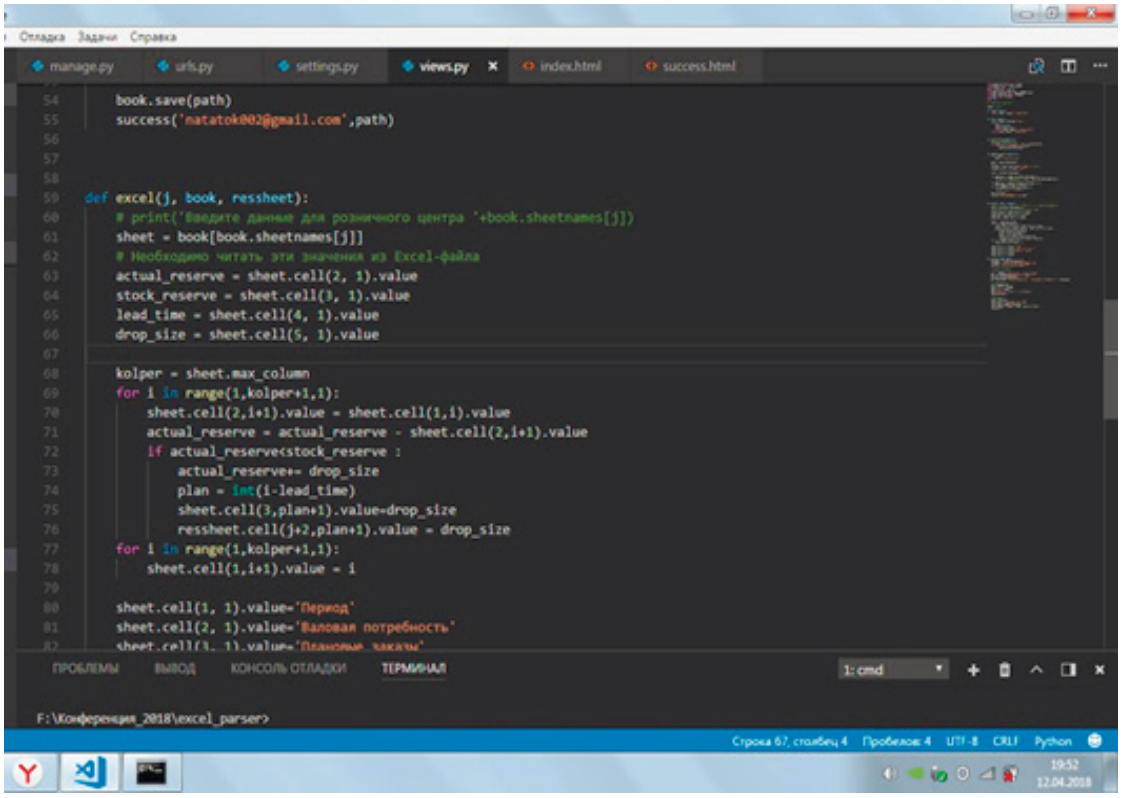

Рис. 1. Обработка загруженного файла

Приложение 4

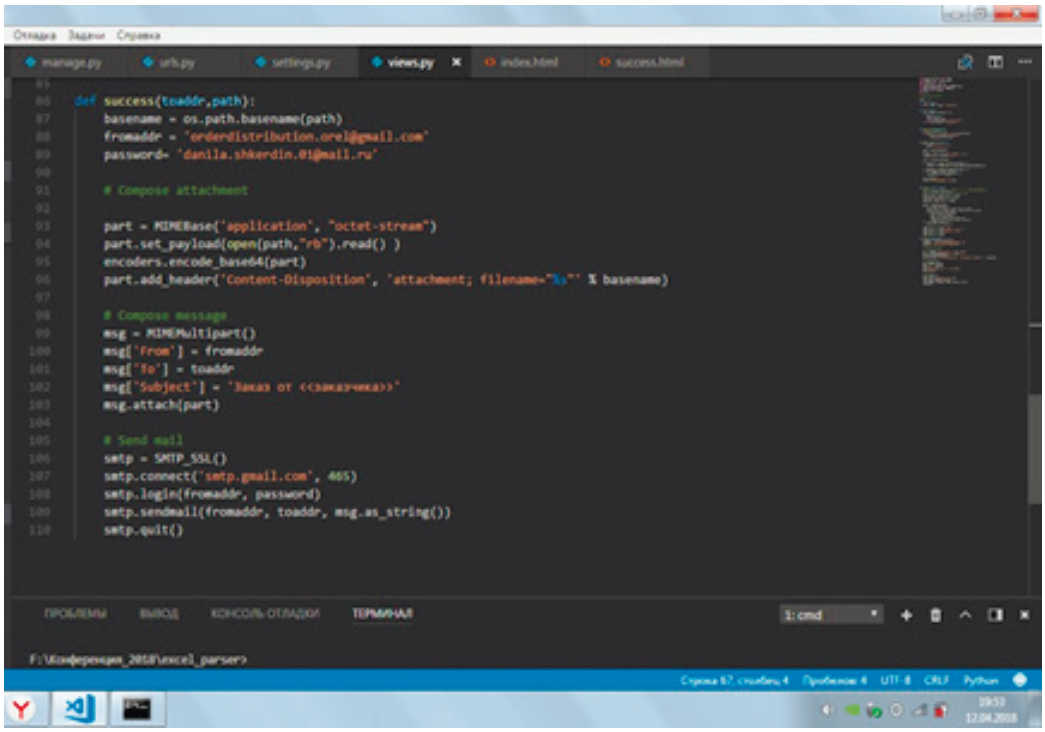

Рис. 1. Блок программы, отвечающий за передачу выходного файла на почту поставщику

Приложение 5

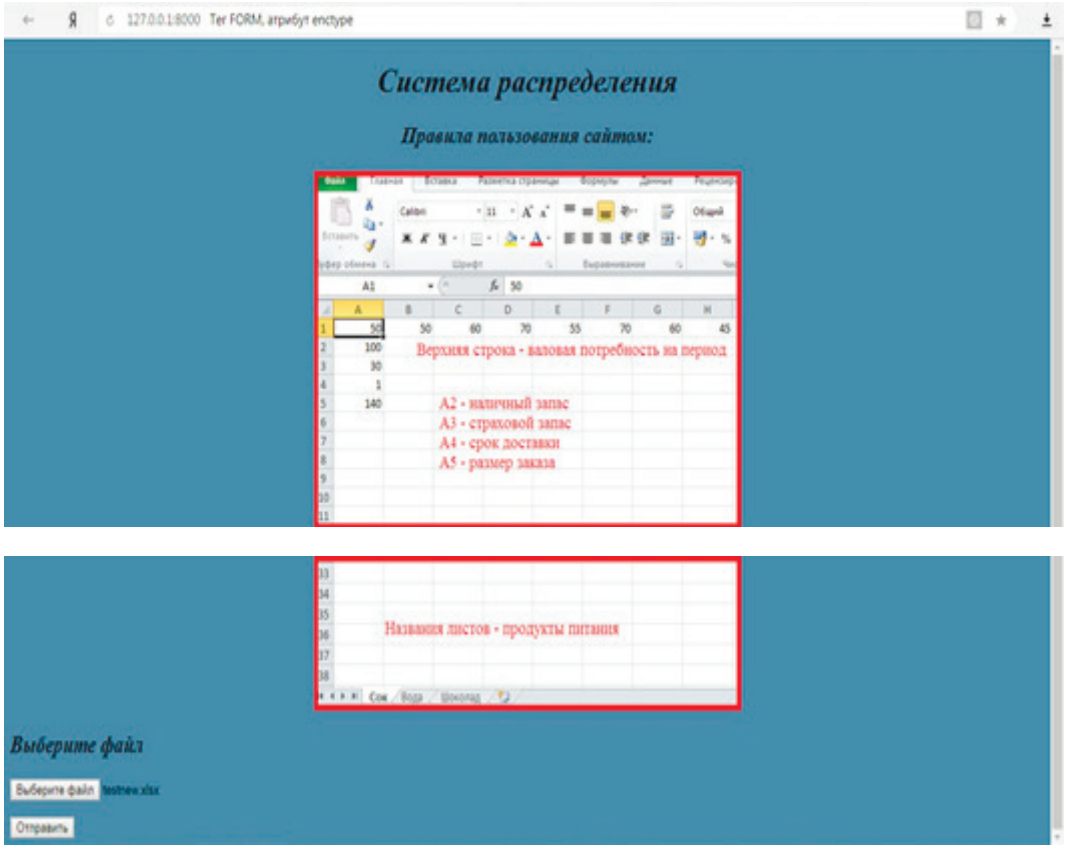

Рис. 1. Внешний вид сайта

Рис. 2. Обновление сайта после загрузки, обработки и отправки файла на e-mail. Ссылка для скачивания файла

Заключение

В процессе выполнения поставленных цели и задач, мы научились работать с фреймворком Django и узнали о возможностях языка программирования Python, сочетающих в себе скорость разработки с возможностью использовать современные инструменты для работы с данными.

Также создали функциональный сайт, ставший логистическим сервисом по работе с документами в формате MS Excel, научились эффективно работать с большими данными.

Список литературы

1. https://python-scripts.com/django-obzor
2. https://djbook.ru/rel1.9/
3. https://habrahabr.ru/company/otus/blog/331998/