Привет!

Спасибо заранее за консультацию. Буду наглеть, скажи😊

Задуманная штука планируется в использовании при оценке размера ущерба (расчет восстановительной стоимости). Основная цель – обучение, задачки уже надоели.

**Схема общего расчета.**

Весь расчет строится в Excel (набор связанных таблиц) и представляет собой следующую последовательность:

* объемы восстановительных работ (на основании результатов осмотра и технической необходимости);
* объем материалов (исходя из набора работ);
* объем допработ (доставка, подъем, мусор, уборка и т.д.);
* **расчет средней арифметической удельной стоимости восстановительных и допработ (это планирую автоматизироать);**
* расчет средней арифметической удельной стоимости материалов;
* калькуляция восстановительных затрат (здесь все сводим и получаем ИТОГО).

**Что и почему хочу автоматизировать.**

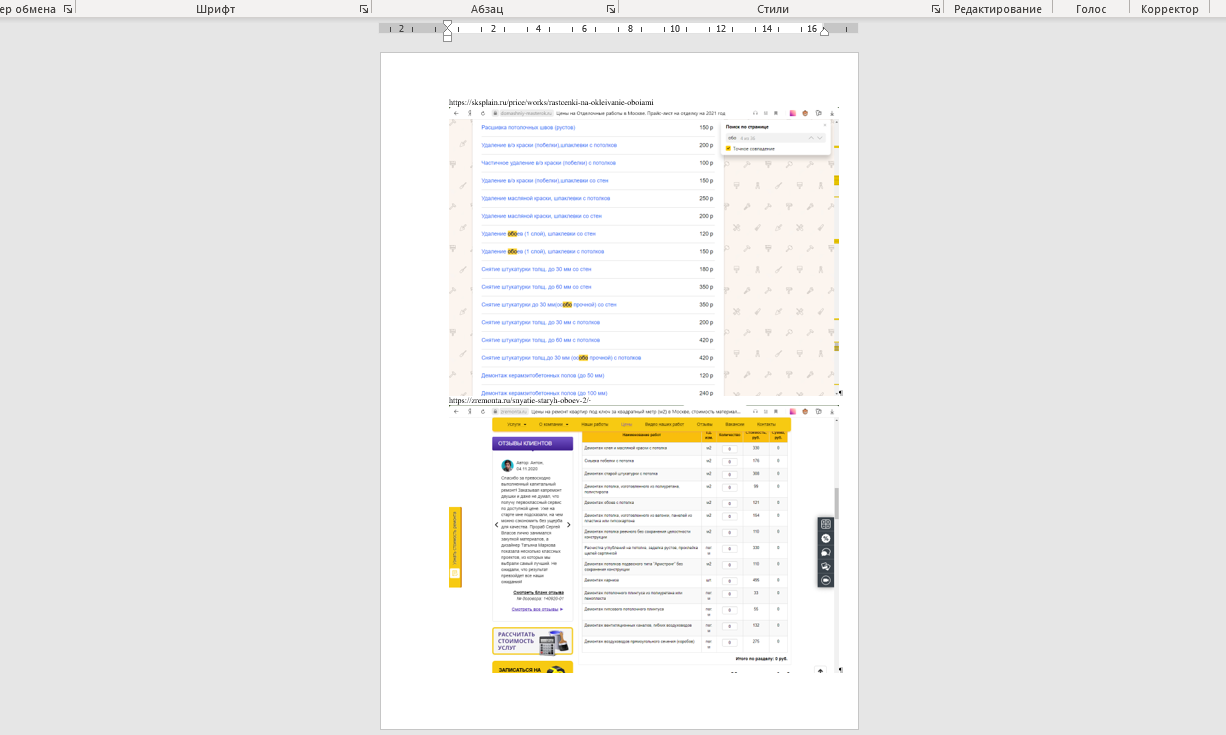
В данный момент это считается в Excel ручным способом и занимает много времени (позиций может быть сотня, на каждую 2-3 минуты, соответственно на рутинную механическую работу может уйти до 15 часов):

* как правило лезем на известные сайты, берем актуальную расценку (проверяем на актуальность если это возможно), вставляем в Excel;
* делаем скриншот прайса (место, где сидит эта работа); потом все скриншоты собираются в один Word-овский документ и печатаются.

Изображение выглядит как стол

Автоматически созданное описание

**Рис. 1.:** *Пример как расчет по средним выглядит в Excel.*



**Рис. 2.:** *Пример как выглядит скриншот в Word (фиксируется скриншот + URL).*

**Как бы я хотел, чтобы это работало:**

* выбирается десяток работ (пока непонятно как это решить в плане интерфейса), например флажком;
* по известным адресам проходит комплексная проверка (наличие-отсутствие страницы, дата если это возможно, наличие-отсутствие работы);
* далее по ним формируется массив (парсим) с основной информацией (наименование работы как это указано на сайте, наименование компании, URL, удельная расценка);
* массив выводится в браузере в графическом виде с возможностью отметки от 1-й до 3-х позиций (то что пойдет в расчет);
* формируется итоговый массив, который выводится в браузер (перед тем как его направить в Excel);
* после этого направляем данные в Excel;
* параллельно с этим формируем в Word скрины прайсов (чтобы в них попадали нужные позиции) с URL;
* на этом работа итерация заканчивается и можно отмечать новый набор.

**Что понятно по вводной информации:**

* есть откорректированный список работ;
* есть ссылки на конкретные прайсы (страницы) конкретных сайтов строительных компаний по 4-6 штук на каждую работу (связь между наименованием работы, страницей и обозначением работы на странице);
* структура страниц везде разная, парсинг (если он нужен) настраивается индивидуально по каждой странице;

**Что хотелось бы представить на первом этапе:**

* структуру всего этого (схема, блоки);
* в чем их можно реализовать;
* насколько это серьезно.

**Что готово:**

* заготовлена часть JSON файлов с данными по работам (исходные данные для доступа к страницам прайсов и работам);
* прописана часть кода на js, которая формирует массив для доступа к прайс-листам.

**Как я укрупненно вижу последовательность действий:**

1. выводим наименование конструктивных элементов (по ним будут выдаваться группы работ): потолки, стены, полы, электрика, сантехника (frontend);
2. отмечаем один конструктивный элемент (frontend);
3. выводим перечень работ по соответствующему конструктивному элементу (frontend);
4. отмечаем необходимую группу работ (frontend);
5. формируем массив (название работы на кириллице, единица измерения, id, данные по аналогам) для отправки на backend (frontend) и проведения процедуры парсинга (frontend);
6. отправляем на backend (frontend);
7. получаем данные на backend (backend);
8. проводим процедуру парсинга (цены или их отсутствие);
9. отправляем результат на frontend (backend);
10. получаем информацию с backend (frontend);
11. выводим результаты в браузер (frontend);
12. отмечаем нужные цены (frontend);
13. формируем массив (данные для построения в Excel и формирования скриншотов);
14. отправляем на backend (frontend);
15. получаем данные на backend (backend);
16. формируем файл Excel (среднее арифметическое) для скачивания (backend);
17. формируем файл Word (скриншоты прайс-листов) для скачивания;
18. отправляем ссылки для скачивания на frontend (backend);
19. получаем информацию с backend (frontend);
20. выводим ссылки в браузер (frontend);
21. скачиваем файлы нажимая на ссылку получаем информацию с backend (frontend).